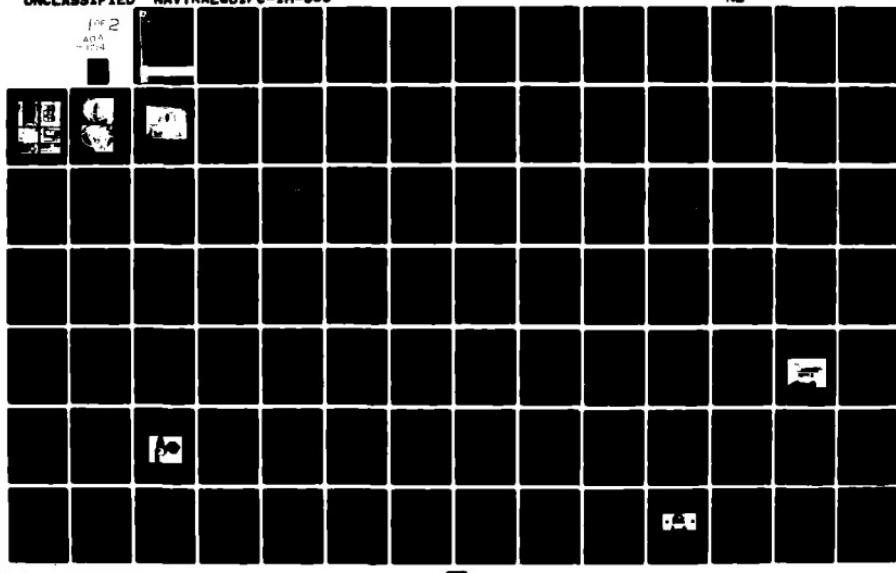


AD-A117 141 NAVAL TRAINING EQUIPMENT CENTER ORLANDO FL
MATH MODEL DESCRIPTION FOR THE VISUAL TECHNOLOGY RESEARCH SIMUL--ETC(U)
JAN 82 E M HOLLER
UNCLASSIFIED NAVTRAEGUIPC-IH-335

F/6 12/1

NL



(12)



TECHNICAL REPORT: NAVTRAEEQUIPCEN IH-335

AD A117141

MATH MODEL DESCRIPTION FOR THE VISUAL
TECHNOLOGY RESEARCH SIMULATOR (VTRS)
CONVENTIONAL TAKEOFF AND LANDING (CTOL)
WEAPON DELIVERY VISUAL SYSTEM

Edward M. Holler
Visual Technology Research Simulator
Naval Training Equipment Center
Orlando, FL 32813

January 1982

Final Report

DoD Distribution Statement
Approved for Public Release
Distribution Unlimited

Naval Air Systems Command
Code 4133J
Washington, D.C. 20361

DTIC
ELECTED
JUL 21 1982

DTIC FILE COPY

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

82 07 21 003

Technical Report: NAVTRAEEQUIPCEN IH-339

MATH MODEL DESCRIPTION FOR THE
VISUAL TECHNOLOGY RESEARCH SIMULATOR (VTRS)
CONVENTIONAL TAKEOFF AND LANDING (CTOL)
WEAPON DELIVERY VISUAL SYSTEM

Edward M. Holler
Code N-732
Advanced Simulation Concepts Laboratory
Naval Training Equipment Center
Orlando, Florida 32813

January 1982

GOVERNMENT RIGHTS IN DATA STATEMENT

Reproduction of this publication in
whole or in part is permitted for any
purpose of the United States Government.

Approved:

DAVID P. GLENN
Director, Research Department

A. M. SPOONER
Head, Advanced Simulation Concepts Laboratory

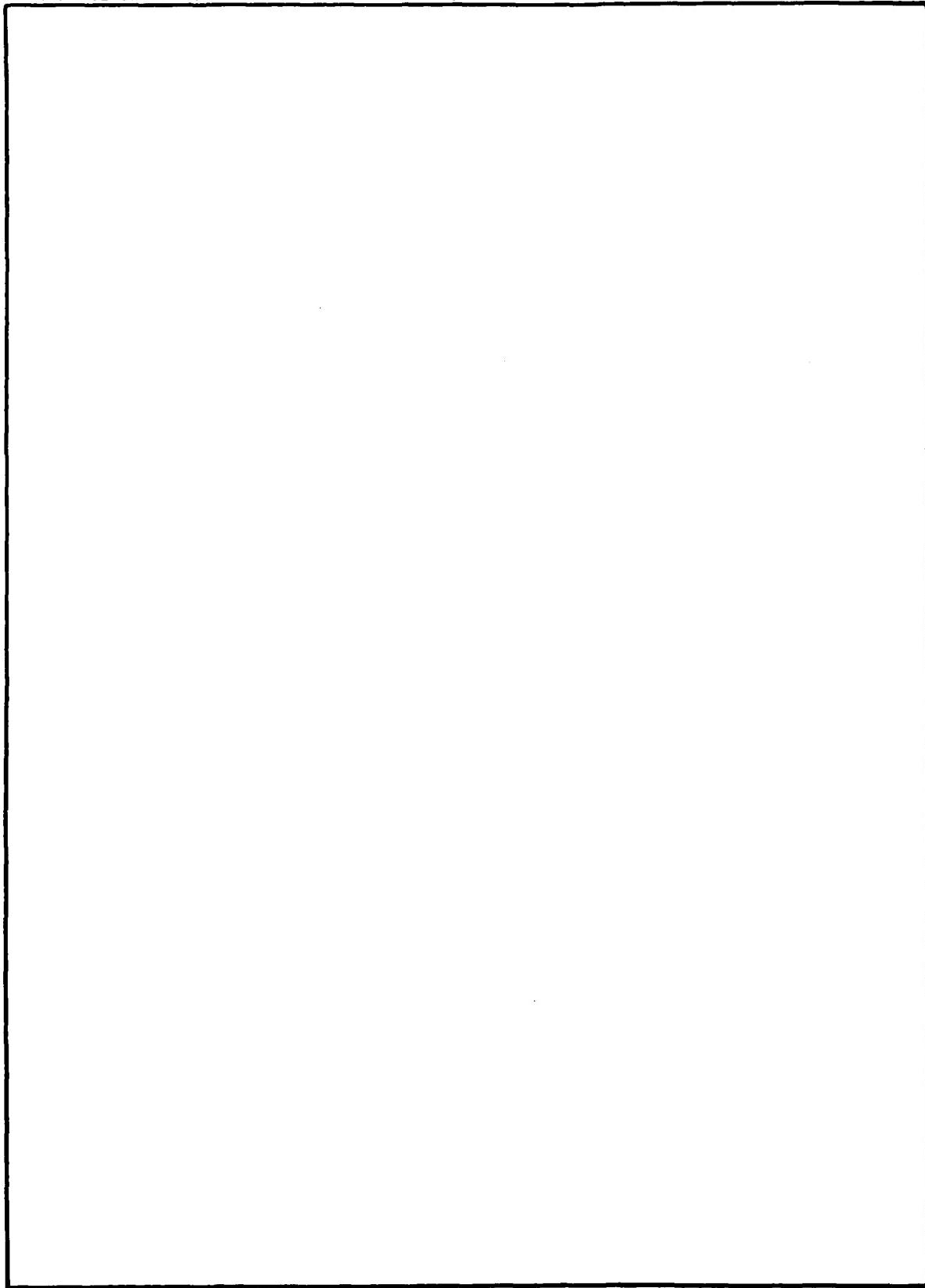
NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NAVTRAEEQUIPCEN IH-335	2. GOVT ACCESSION NO. Y / 11 111	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Math Model Description for the Visual Technology Research Simulator (VTRS) Conventional Take Off and Landing (CTOL) Weapon Delivery Visual System	5. TYPE OF REPORT & PERIOD COVERED Final	
7. AUTHOR(s) Edward M. Holler	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Visual Technology Research Simulator Naval Training Equipment Center Orlando, FL 32813	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE63733N W1200-RM 0785-3	
11. CONTROLLING OFFICE NAME AND ADDRESS Commanding Officer Naval Training Equipment Center Orlando, FL 32813	12. REPORT DATE January 1982	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Air Systems Command Code 4133J Washington, D.C. 20361	13. NUMBER OF PAGES 134	
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; distribution unlimited.	15. SECURITY CLASS. (of this report) Unclassified	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Visual Technology Research Simulator Math Model Flight Simulation Weapon Delivery Simulation		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report provides a technical description of the software for weapon delivery simulation for the T-2C aircraft at the Visual Technology Research Simulator (VTRS) facility. A summary of the complete system is provided first and then each of the software modules is described in detail. System flow charts, axis system diagrams, math model vector diagrams, and logic tables are provided. The mathematical equations are done in Fortran. Pictures of visual scenes and moving models are also provided to illustrate system capability.		

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

A large, empty rectangular box occupies most of the page, indicating a significant portion of the document has been redacted or removed.

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

NAVTRAEEQUIPCEN IH-335

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
I	INTRODUCTION.	7
	FLIGHT AND VISUAL DISPLAY EQUIPMENT	7
	COMPUTER IMAGE GENERATION EQUIPMENT	8
	VTRS CTOL SYSTEM.	8
	VISUAL SOFTWARE OVERVIEW.	13
	MODULES	13
	SUBROUTINES	13
	VISUAL MODULE/SUBROUTINE SUMMARY.	15
	FLIGHT/VISUAL INTERFACE	15
	WINDOW/POINT OF INTEREST.	15
	TARGET IMAGE PROJECTOR.	15
	MOVING MODEL CONTROL.	15
	WEAPON SCORING.	15
	VISUAL/CIG INTERFACE.	15
	MOVING MODEL SUBROUTINES.	16
II	FLIGHT/VISUAL INTERFACE MODULE.	17
	FLIGHT/VISUAL INTERFACE MODULE FUNCTIONAL DESCRIPTION	17
	FLIGHT/VISUAL FUNCTIONS	19
III	WINDOW/POINT OF INTEREST MODULE	27
	WINDOW/POINT OF INTEREST FUNCTIONAL DESCRIPTION	27
IV	TARGET IMAGE PROJECTOR.	34
	TARGET IMAGE PROJECTOR MODULE FUNCTIONAL DESCRIPTION.	34
V	MOVING MODEL CONTROL.	42
	MOVING MODEL CONTROL MODULE FUNCTIONAL DESCRIPTION.	42
VI	WEAPON SCORING.	46
	WEAPON SCORING MODULE FUNCTIONAL DESCRIPTION.	46
VII	VISUAL/CIG INTERFACE.	52
	VISUAL/CIG INTERFACE MODULE FUNCTIONAL DESCRIPTION.	52

APPENDICES

<u>Appendix</u>		<u>Page</u>
A	AIRCRAFT CARRIER DRIVER	65
	CARRIER DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION.	66
B	FLAK DRIVER SUBROUTINE.	69
	FLAK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION	70
C	FLOLS DRIVER.	79
	FLOLS DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION.	79
D	GENERAL PURPOSE DRIVER.	89
	"GENERAL" DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION.	89
E	AIR-TO-SURFACE ROCKET DRIVER SUBROUTINE	93
	ROCKET DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION	93

NAVTRAEEQUIPCEN IH-335

APPENDICES (CONTINUED)

	<u>Page</u>
F SURFACE-TO-AIR MISSILE (SAM) DRIVER SUBROUTINE	97
SAM DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION	97
G TA4J DRIVER SUBROUTINE	107
TA4J DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION	108
H TANK DRIVER	111
TANK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION	111
I TRUCK DRIVER	123
TRUCK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION	123

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unnumbered	<input type="checkbox"/>
Journal Article	<input type="checkbox"/>
<hr/>	
10	
20	
30	
40	
50	
60	
70	
80	
90	
100	
110	
120	
130	
140	
150	
160	
170	
180	
190	
200	
210	
220	
230	
240	
250	
260	
270	
280	
290	
300	
310	
320	
330	
340	
350	
360	
370	
380	
390	
400	
410	
420	
430	
440	
450	
460	
470	
480	
490	
500	
510	
520	
530	
540	
550	
560	
570	
580	
590	
600	
610	
620	
630	
640	
650	
660	
670	
680	
690	
700	
710	
720	
730	
740	
750	
760	
770	
780	
790	
800	
810	
820	
830	
840	
850	
860	
870	
880	
890	
900	
910	
920	
930	
940	
950	
960	
970	
980	
990	



NAVTRAEEQUIPCEN IH-335

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	VTRS CTOL System	9
2	CTOL Hardware Configuration	10
3	CTOL Cockpit and Dome	11
4	CTOL Projector System	12
5	Weapon Visual Software Flowchart	14
6	Flight Earth Axis System	17
7	CIG Earth Axis System	18
8	Body Axis System	19
9	Relationship Between Aircraft Body Axis and Eyepoint Axis Systems	22
10	Target Axis System	24
11	Point of Interest in Observer Axis	27
12	Point of Interest (POI) Offset in Observer Axis	28
13	Point of Interest in CIG Earth Axis System	29
14	Point of Interest Angles in Observer Axis	30
15	Target Image Projector Position Relative to Eyepoint, Side View	34
16	Screen Intersection Vectors in Display Axis System	35
17	The Azimuth Vectors in the Tip Axis System	38
18	Tip Azimuth Angles in Tip Axis System, Top View	39
19	Vector Diagram for Bomb Miss Distance	47
20	Vector Diagram for Rocket/Bullet Miss Distance	47
21	Vector Diagram for CPA Values Below the Ground	51
A-1	Carrier Landing Approach on VTRS System	65
A-2	Carrier Reference Point and Center of Gravity Definition in Carrier Axis System	66
B-1	Anti-Aircraft FLAK Burst	69
B-2	Location of FLAK and SAM Launch Pad in CIG Earth Axis	70
B-3	FLAK Burst Position in CIG Earth Axis System	73
C-1	FLOLS Position on Carrier in Carrier Axis System	80
C-2	FLOLS Position and Attitude for Meridian Airport Data Base	81
C-3	FLOLS Meatball Position Relative to Aircraft Glideslope	84
C-4	FLOLS Datum Line and Meatball Position, Side View	86
E-1	Air-to-Surface Missiles (2.75" Rockets)	93
E-2	Rocket Centroid Definition in Rocket Axis System	94
F-1	Surface-to-Air Missile (SAM) Launch	97
F-2	SAM Launch Pad Setup, CIG Earth Axis System	98
F-3	SAM Trajectory Vectors, CIG Earth Axis System	100
G-1	TA4J in Formation Flight	107
H-1	T-62 Tank	111
H-2	Tank Route on Gunnery Range	112
H-3	Tank Route on Twin Towns	115
H-4	Tank Route in River Valley	118

NAVTRAEEQUIPCEN IH-335

LIST OF ILLUSTRATIONS (CONTINUED)

<u>Figure</u>		<u>Page</u>
I-1	Truck Convoy.	123
I-2	Truck Convoy Route on Gunnery Range	124
I-3	Truck Convoy Route in Twin Towns.	126
I-4	Truck Convoy in River Valley.	130

NAVTRAEEQUIPCEN IH-335

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	MOVING MODEL OPTIONS FOR DATABASE ENVIRONMENTS.	43
2	MOVING MODEL CONTROL FLAGS.	43
3(A)	SEL TO DEC INTERFACE FLOATING POINT PARAMETERS 32 BIT WORDS.	54
3(B)	SEL TO DEC INTERFACE CONTROL FLAGS.	58
4	DEC TO SEL INTERFACE (A) FRAME RATE BUFFER.	61
5	DEC TO SEL INTERFACE (B) DATA BASE INITIALIZATION BUFFER. . .	62

SECTION I

INTRODUCTION

This math model report describes the visual software for the Conventional Take-off and Landing (CTOL) system at the Visual Technology Research Simulator (VTRS) facility. The VTRS CTOL system is a real-time flight simulator for the Navy T2C jet aircraft. This system includes a six degree of freedom motion platform, a G-seat, T2C cockpit, an experimenter operator station, and a full color wide angle visual system. Capabilities of this system include aircraft carrier takeoff and landings, airport takeoff and landings, formation flight, and air-to-ground combat. The air-to-ground capabilities include 2.75 inch rockets, 50 caliber machine guns and bombs which can be launched against six moving targets and eight fixed targets. Also included is anti-aircraft flak and surface-to-air missiles which can be launched against the T2C. Hits are scored and displayed for fixed and moving targets and against the pilot.

The visual software referenced in this report, except for Singer-Link executive, target projector routines, and utility programs, was designed, coded, implemented, and documented at the Naval Training Equipment Center (NTEC) as an in-house effort. The original system delivered to NTEC included a Singer-Link wide angle model board visual system and a T2C flight simulator that was limited to aircraft carrier takeoff and landings. NTEC later purchased a General Electric Computer Image Generation (CIG) system and modified the above Singer-Link model board flight simulator so that CIG images could be displayed to the pilot in the dome. Both the GE CIG system and the flight simulator were recently upgraded to permit research in the air-to-ground area. The CIG system upgrade was performed by GE and the flight simulator system upgrade was done by NTEC. This report addresses the new visual air-to-ground software that was developed by NTEC.

The hardware configuration for the VTRS CTOL simulator is illustrated in Figure 1. Note that the simulator is divided into two basic systems: (1) the flight/visual system; and (2) the Computer Image Generation (CIG) system. Photographs of the VTRS CTOL hardware are provided in Figures 2, 3 and 4. This equipment is summarized below:

FLIGHT AND VISUAL DISPLAY EQUIPMENT.

- a. Flight cockpit, G-seat, and motion platform.
- b. Visual screen (dome), and two-channel projection system.
 - (1) Target channel has a high resolution servo-driven projection system for target image display.
 - (2) Background channel has a low resolution wide angle fixed projection system for background scene display.

c. Experimenter operator station (EOS) to control flight and visual functions, initiate data recording, monitor pilot performance, etc.

d. High speed analog/digital interface and computer system drives the simulator and interfaces with the Computer Image Generation (CIG) system. The computer system includes three System Engineering Laboratories (SEL) 32/75 minicomputers, memory, and two control data 9762 eighty mega byte discs.

(1) Central Processing Unit (CPU) 1 is the master computer for the visual and flight computers and drives the EOS.

(2) CPU 2 receives data from flight, drives the visual projectors and inputs visual parameters to CIG for visual displays.

(3) CPU 3 receives pilot parameters from cockpit and computes flight parameters for motion, G-seat, and visual computer.

COMPUTER IMAGE GENERATION EQUIPMENT.

a. Frame I equipment includes two DEC PDP 11/T55 computers and four RK05J disc units. One PDP 11/T55 interfaces with the SEL computers and provides inputs for Frame II equipment.

b. Frame II equipment is digital image processing hardware that converts three dimensional data into two dimensional data and performs distortion correction for images so that they will appear undistorted in the dome.

c. Frame III equipment is digital raster processing hardware that inputs Frame II data and puts it in raster format for display in the dome through the background and target projectors.

VTRS CTOL SYSTEM.

An overview of the VTRS CTOL system operation is now provided. The simulator's initial flight conditions, visual functions, CIG environment requests, data monitoring and data recording are controlled from the EOS, which is the primary control device for the simulator. The SEL executive computer processes EOS requests and is the master for the flight and visual SEL computers. The SEL flight computer processes pilot cockpit responses, computes the appropriate flight equations, and drives the cockpit, G-seat, and motion platform. The SEL visual computer inputs aircraft position, attitude, and other pertinent data from flight and computes the appropriate drive parameters for the visual projection hardware in the dome. The CIG system inputs all viewpoint and moving model positions and attitudes, environment requests, etc., from the SEL visual computer. The CIG Frame I computers process these parameters for input to the CIG Frame II hardware which converts three dimensional data into two dimensional data and performs distortion correction on this data so visual pictures will appear undistorted in the dome. Frame III equipment inputs the data from Frame II and puts it in raster

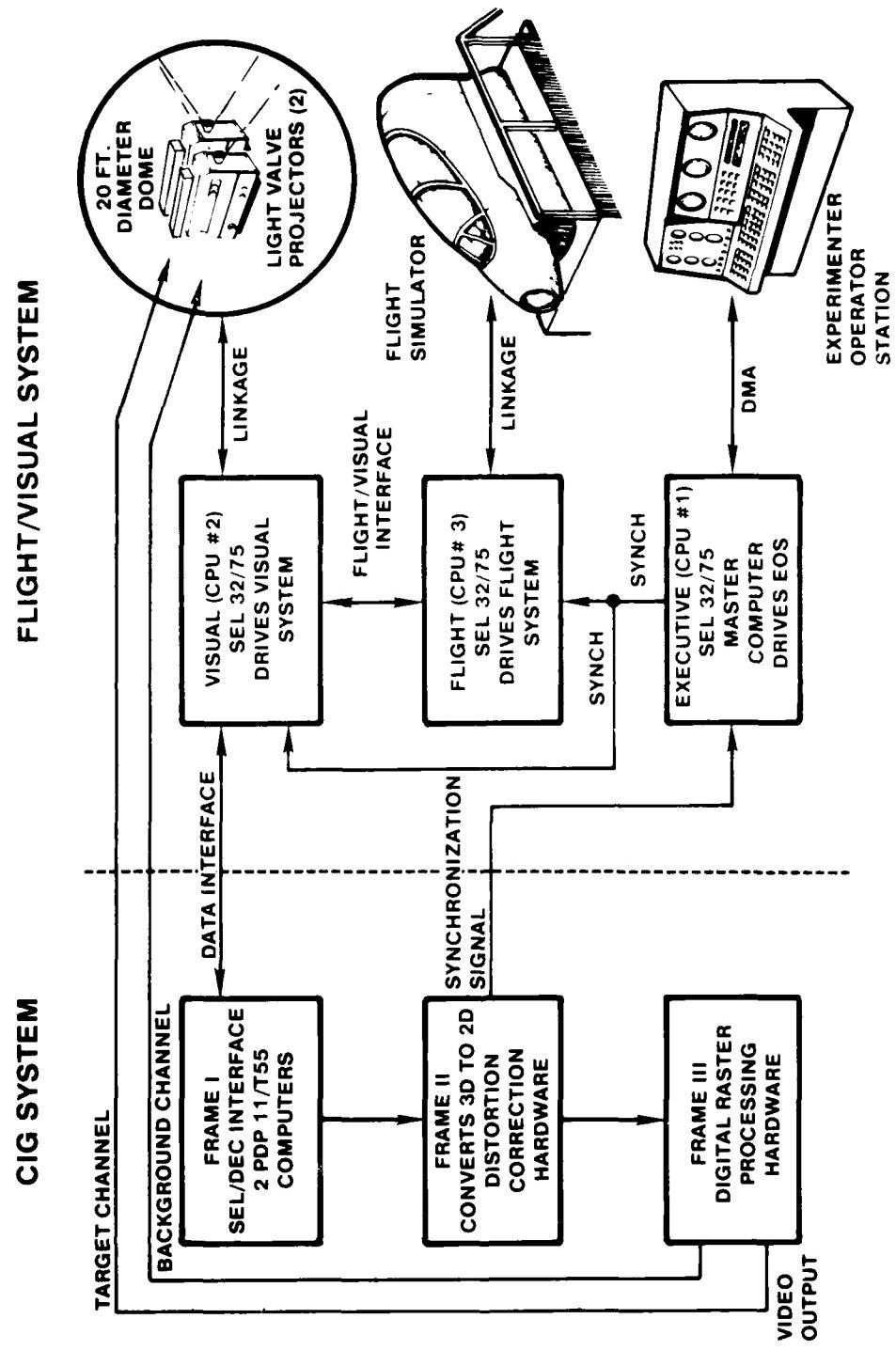


Figure 1. VTRS CTOL System.

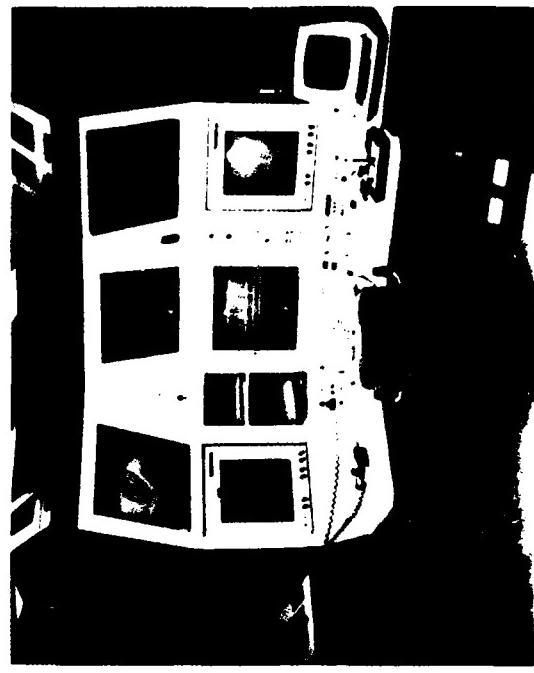
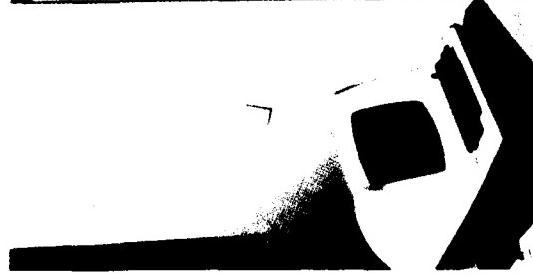
NAVTRAEEQUIPCEN IH-335



Frame II and III. CIG Hardware.



Frame I. CIG Computer System.



Executive, Flight and Visual Computer System. Experimenter Operator Station.

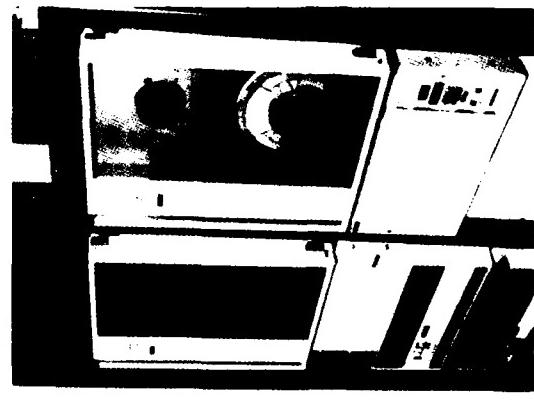
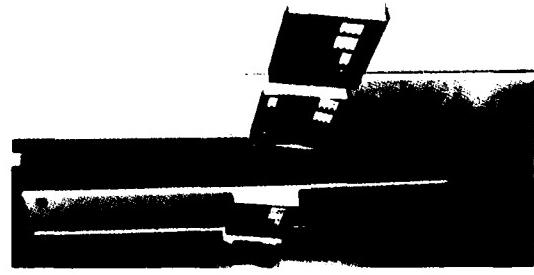
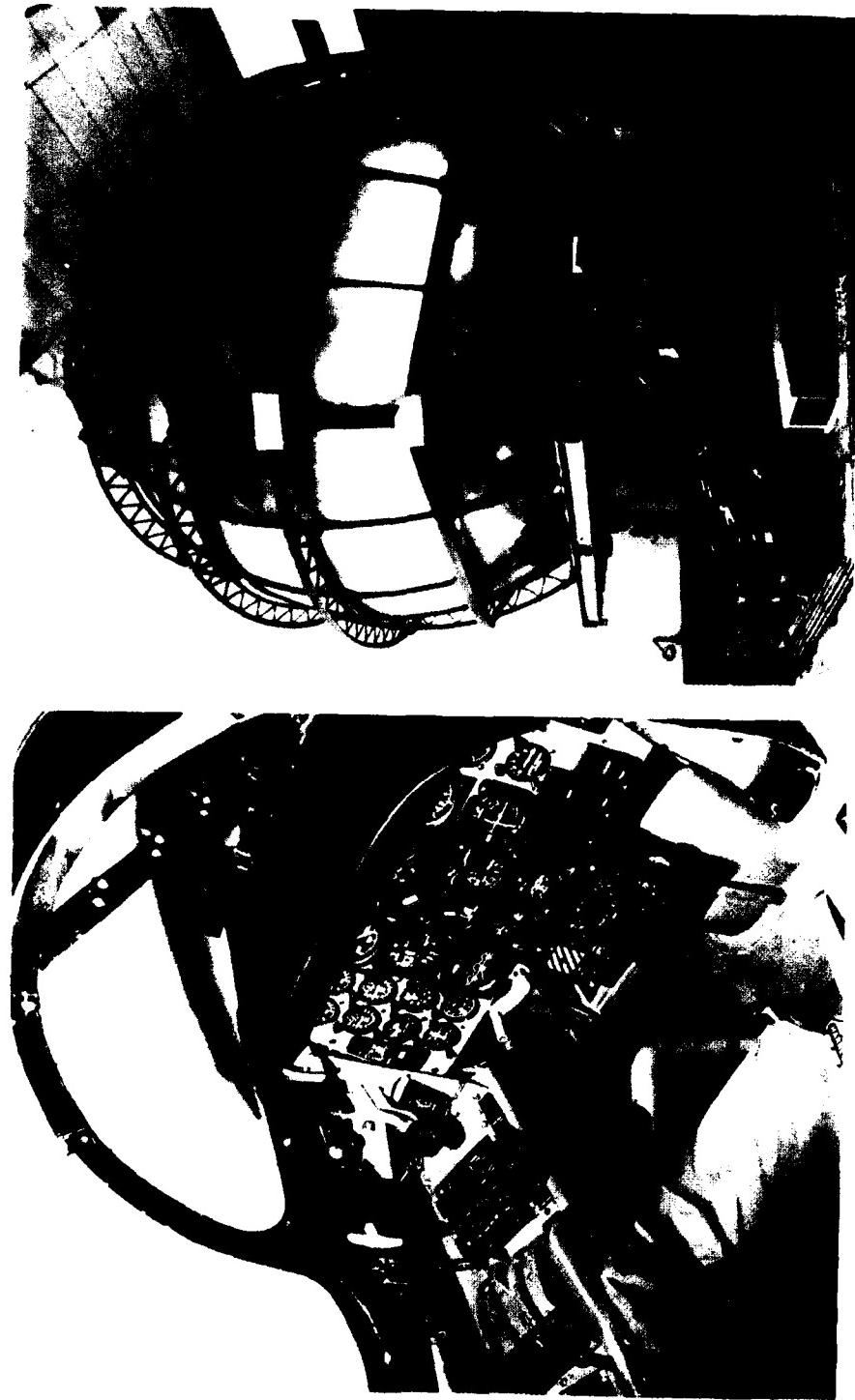


Figure 2. CTOL Hardware Configuration



NAVTRAEEQUIPCEN IH-335



Flight Simulator Cockpit and G Seat Visual System Dome and Motion Platform

Figure 3. CTOL Cockpit and Dome.

NAVTRAEEQIPCEN IH-335

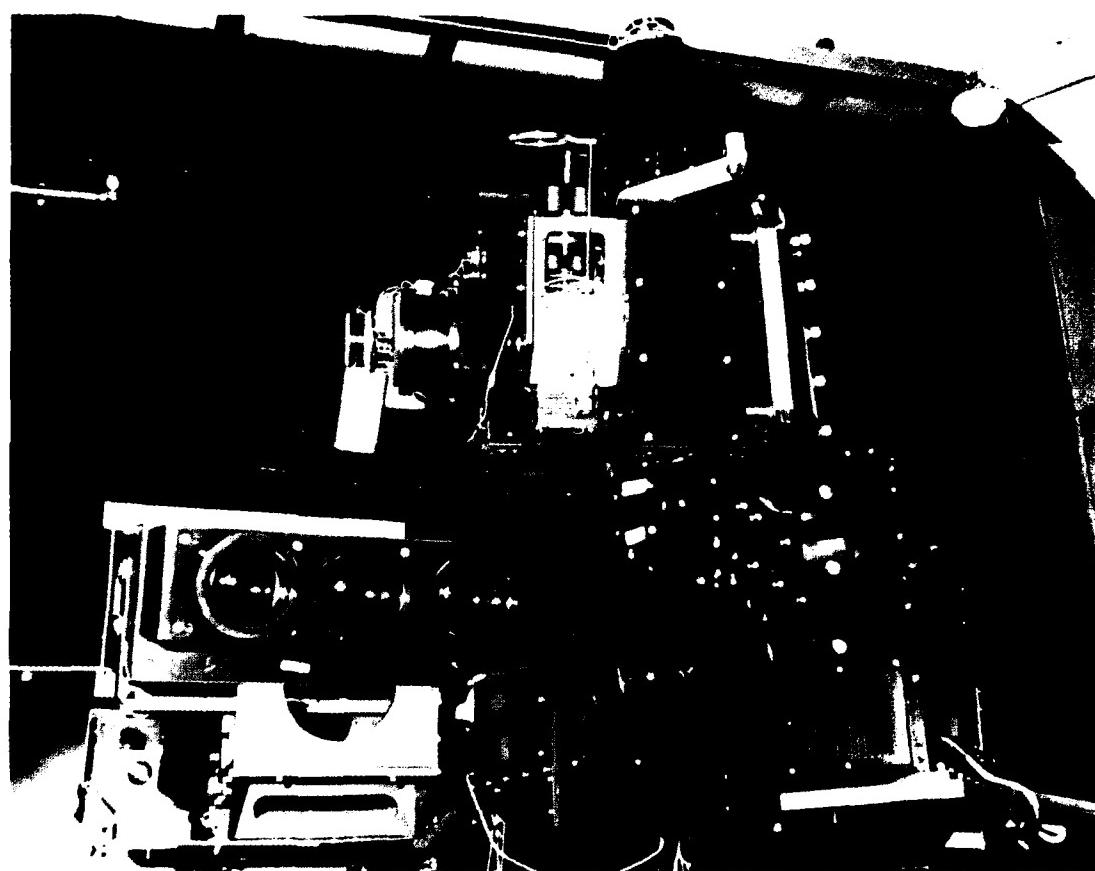


Figure 4. CTOL Projector System.

NAVTRAEEQUIPCEN IH-335

format for the background and target projectors in the dome. For a more complete explanation of the computer system, refer to Technical Note: NAVTRAEEQUIPCEN IH-329, "Visual Technology Research Simulator Computer Programmer's Guide," December 1980. The remainder of this report is a math model description of the visual software that executes in the SEL visual computer referenced above. For more details on the CIG system, refer to (1) "Visual Technology Research System Computer Generated Imagery System," Trainer Programming and Math Model Report, Volume I, June 1981; and (2) "System Description, Aviation Wide-Angle Visual System, Computer Image Generator Visual System," (revised May 1981).

VISUAL SOFTWARE OVERVIEW.

The visual software contains six software modules and nine subroutines. All modules and subroutines are written in Fortran. The software modules are called from the visual executive program and execute at a thirty Hertz rate. The visual executive program is written in SEL assembly language. The subroutines are called by the software modules at a thirty Hertz rate, but only when requested by the specified visual task. This significantly reduces visual program execution time. The weapon visual software flow chart is shown in Figure 5 and is broken down as follows:

MODULES.

- a. ZM6SW#33: Flight/Visual Interface
- b. ZM6SW#36: Window/Point of Interest
- c. ZM6SW#37: Target Image Projector
- d. ZM6SMOVE: Moving Model Control
- e. ZM6SCORE: Weapon Scoring
- f. ZM6SWCGI: Visual/CIG Interface

SUBROUTINES.

- a. WCARRIER: Carrier Driver
- b. WFLAK : FLAK Driver
- c. WFLOLS : FLOLS Driver
- d. WGENERAL: General Purpose Driver
- e. WROCKETS: Rocket Driver
- f. WSAM : Sam Driver

NAVTRAEEQUIPCEN IH-335

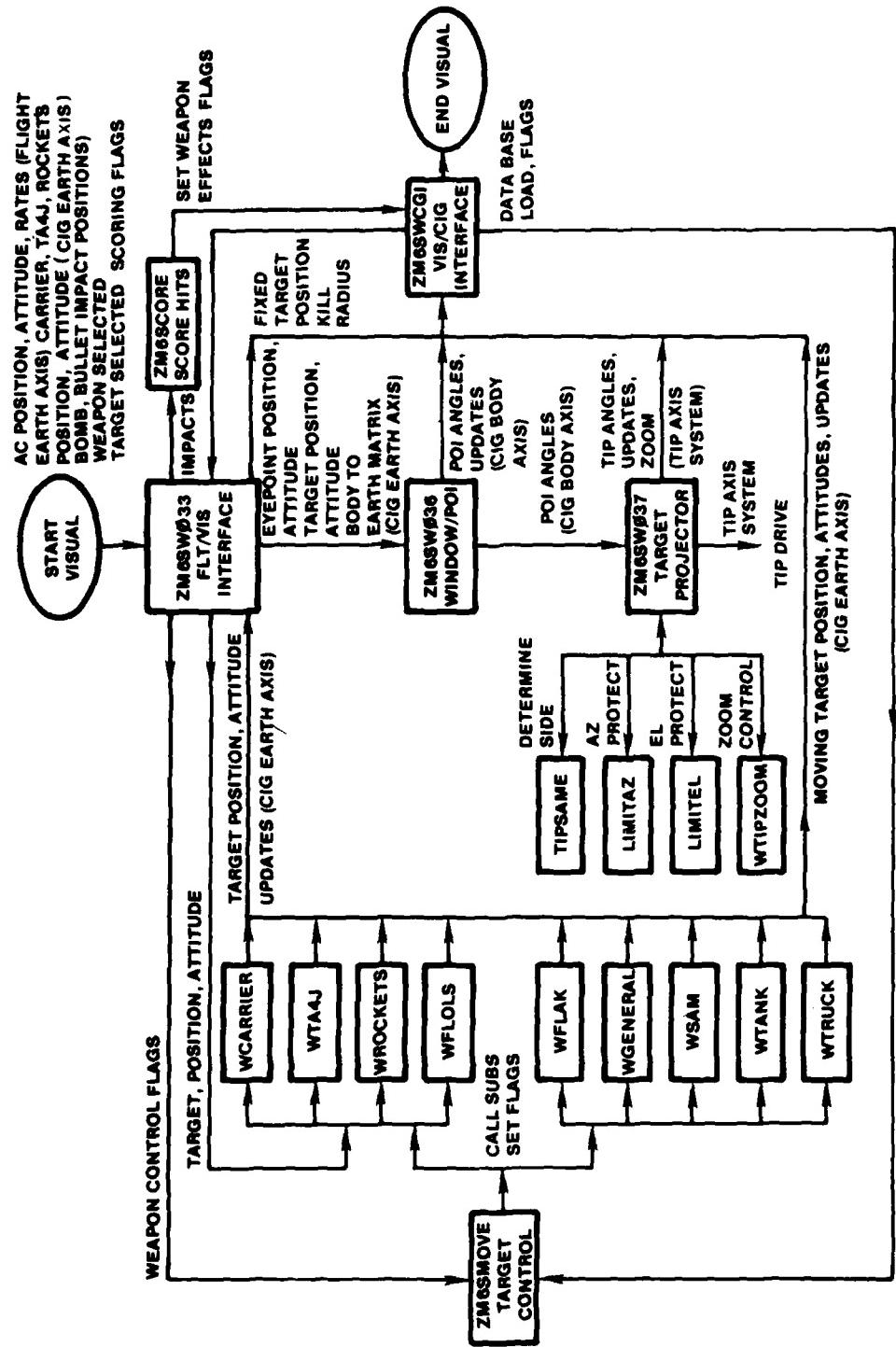


Figure 5. Weapon Visual Software Flowchart.

- g. WTA4J : TA4J Driver
- h. WTANK : Tank Driver
- i. WTRUCK : Truck Driver

VISUAL MODULE/SUBROUTINE SUMMARY.

The visual software is structured so that each module and subroutine performs a specific function. These functions are summarized below and a detail description is provided in Section II. All mathematical equations in this report are shown in Fortran IV format.

FLIGHT/VISUAL INTERFACE. This module interfaces VTRS flight and visual parameters. Flight parameters are computed by the flight computer (CPU #3) and passed via datapool to the visual computer (CPU #2). This module converts all flight parameters from the Flight Axis System to the CIG Axis System for visual processing.

WINDOW/POINT OF INTEREST. This module computes azimuth and elevation angles for the observer point of interest in the Observer Coordinate System. The point of interest is typically a target such as an aircraft carrier or tank. This module defines the window through which the observer looks to see this target.

TARGET IMAGE PROJECTOR. This module inputs the azimuth and elevation angles from Window/Point of Interest module and converts them into the Target Image Projector coordinates. These angles are then used to drive the target projector servos in the dome. This puts the target in the proper position in the dome.

MOVING MODEL CONTROL. This module looks at the loaded database and then activates the appropriate subroutine to drive the models that exist in that environment. Each database has its own unique models that must be driven by the weapons visual software. For example, in the aircraft carrier database, this module would activate the carrier and Fresnel Lens Optical Landing System (FOLDS) moving models.

WEAPON SCORING. This module scores weapon deliveries of bombs, bullets, and rockets shot from the T2C aircraft. Bombs are scored as ground miss distances. Rockets and bullets are scored as Closest Point of Approach (CPA) above the ground. If hits are scored on the tank or truck, this module sends a kill flag to the CIG system, which will display an explosion on the target.

VISUAL/CIG INTERFACE. This module gets the weapon visual parameters, converts them to PDP 11 computer format as required, and stores them in datapool for transfer to the CIG system. These parameters are used to drive the viewpoint and moving model position and attitudes in the CIG system. The CIG database

NAVTRAEEQUIPCEN IH-335

parameters from the CIG system are assigned to appropriate weapon visual parameters. These parameters are used to determine which database is loaded and which moving models should be activated.

MOVING MODEL SUBROUTINES. The nine moving model subroutines are listed in Appendixes A-I. These routines are called by the moving model control module. The function of these routines is to drive the CIG models as required by each database. For example, the Truck Driver routine must drive the truck on the road in the specified database. There are currently three different roadways in each of the three databases with trucks. This report includes a detailed section for each of these subroutines.

SECTION II

FLIGHT/VISUAL INTERFACE MODULE

This module interfaces flight parameters and visual parameters. All flight coordinates are converted from Flight to CIG coordinates in this module. All weapon visual parameters are in the CIG Coordinate System.

FLIGHT/VISUAL INTERFACE MODULE FUNCTIONAL DESCRIPTION.

The coordinate systems are the reference system through which the visual and flight world are defined. A complete understanding of these axis systems is required to comprehend the mathematical description in this report. The following coordinate descriptions should be thoroughly understood before proceeding any further in this section.

The Flight Earth Axis System is shown in Figure 6. It is a standard right-handed rectangular coordinate system with its axis and angles defined below. The preferred order of rotation is yaw, pitch, and then roll. The axis system is summarized as follows:

- a. +X is pointed north.
- b. +Y is pointed east.
- c. +Z is pointed down.
- d. + yaw is defined as a right-hand rotation about the Z axis.
- e. + pitch is defined as a right-hand rotation about the Y axis.
- f. + roll is defined as a right-hand rotation about the X axis.

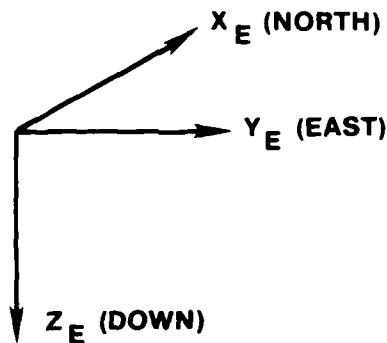


Figure 6. Flight Earth Axis System.

The CIG Earth Axis System is shown in Figure 7. Its axis system differs from the Flight system as follows: (1) The X and Y axis are interchanged; (2) the Z axis is in the opposite direction. The preferred order of rotation of this axis system is also yaw, pitch, roll. This axis system is summarized as follows:

- a. +Y is pointed north.
- b. +X is pointed east.
- c. +Z is pointed up.
- d. + yaw is defined as a right-hand rotation about the Z axis.
- e. + pitch is defined as a right-hand rotation about the X axis.
- f. + roll is defined as a right-hand rotation about the Y axis.

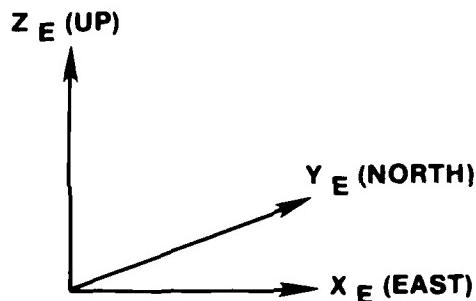


Figure 7. CIG Earth Axis System.

The body axis for both the Flight and CIG systems is shown in Figure 8. These axis systems are summarized as follows:

- a. +X is out the nose of the aircraft.
- b. +Y is out the right wing.
- c. +Z is down.
- d. + azimuth is rotated about the Z axis and to the right.
- e. + elevation is rotated about the Y axis (nose up).
- f. + roll is rotated about the X axis and to the right (right wing down).

The Target Axis System is exactly the same as the two body axis systems described above and is also shown in Figure 8.

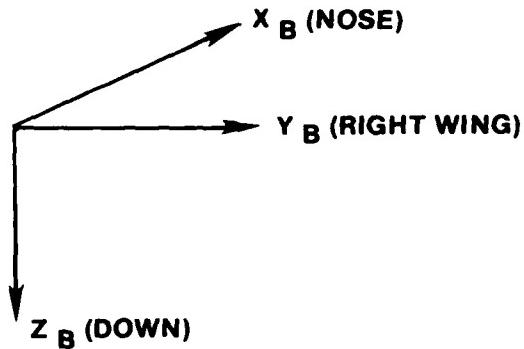


Figure 8. Body Axis System.

FLIGHT/VISUAL FUNCTIONS.

The functional tasks of the Flight/Visual Interface Module are now described in detail.

The nine Flight direction cosines that define the body to earth transfer matrix are computed using a heading correction to compensate for the curvature of the earth. This is needed because the Visual Coordinate System is rectangular, while the earth is obviously spherical. The derivation is shown below:

The heading angle PSI is computed from flight inputs: (1) sine and cosine of heading; (2) angular heading bias resulting from the convergence of longitude as the aircraft nears the north or south poles.

$$\begin{aligned} \text{PSI} &= \text{ATAN}(\text{AFSPSI}/\text{AFCPSI}) + \text{ANGBIAS} \\ \text{CPSI} &= \text{COS}(\text{PSI}) \\ \text{SPSI} &= \text{SIN}(\text{PSI}) \end{aligned}$$

Where:
 PSI is aircraft true heading in degrees
 AFSPSI, AFCPSI are sine and cosine of aircraft heading without angular heading bias
 ANGBIAS is aircraft angular heading bias in degrees
 CPSI is cosine of PSI
 SPSI is the sine of PSI

NAVTRAEEQUIPCEN IH-335

Next, the nine direction cosines that describe the axis system in Figure 6 are derived using the standard Euler angles PSI, THETA, and PHI:

```
AVBEFL11=AFCTHT*CPSI  
AVBEFL21=AFCTHT*SPSI  
AVBEFL31=-AFSTHT  
AVBELF12=AFCPHI*AFSTHT*CPSI - AFCPHI*SPSI  
AVBELF22=AFCPHI*CPSI + AFSPHI*AFSTHT*SPSI  
AVBELF32=AFSPHI*AFCTHT  
AVBELF13=AFSPHI*SPSI + AFCPHI*AFSTHT*CPSI  
AVBELF23=AFCPHI*AFSTHT*SPSI - AFSPHI*CPSI  
AVDELFL33=AFCPHI*AFCTHT
```

Where: AVBEFL11-33 are the nine direction cosines which describe the orientation of the Flight Earth Axis System in Figure 6 with respect to the body axis in Figure 8
AFCTHT, AFSTHT are sine and cosine of aircraft pitch
AFSPHI, AFCPHI are sine and cosine of aircraft roll
CPSI, SPSI are sine and cosine of true aircraft heading

Next, the nine Flight direction cosines are converted into CIG direction cosines. By inspection of Figure 6 and 7, it is obvious that the X and Y axis are interchanged and Z axis is opposite between the two systems. Hence, to convert from Flight to CIG, the X and Y matrix components are swapped and the Z matrix components are negated.

Convert Flight body to earth direction cosines to CIG body to earth direction cosines:

```
AVBDER11=AVBEFL21  
AVBDER21=AVBEFL11  
AVBDER31=-AVBEFL31  
AVBDER12=AVBEFL22  
AVBDER22=AVBEFL12  
AVBDER32=-AVBEFL32  
AVBDER13=AVBEFL23  
AVBDER23=AVBEFL13  
AVBDER33=-AVBEFL33
```

Where: AVBDER11-AVBDER33 are CIG body to earth direction cosines
AVBEFL11-AVBEFL33 are Flight body to earth direction cosines

Since the aircraft position and attitude parameters in flight are defined in the Aircraft Body Axis System, which has its origin at the aircraft center of gravity (C.G.), the visual software must define the pilot's eyepoint position in the Eyepoint Axis System to get the correct picture in the dome. The Aircraft Body Axis and the Eyepoint Axis System have their X, Y, Z axis aligned, but are offset in position by the relationship shown in Figure 9. The translation from the Aircraft Body Axis to the Eyepoint Axis System is derived below:

NAVTRAEEQUIPCEN IH-335

The eyepoint position is computed relative to Aircraft Body Axis System (see Figure 9). Note that the aircraft body axis shifts as the C.G. changes with aircraft weight changes, but the eyepoint position remains fixed relative to the aircraft fuselage. This point is illustrated in Figure 8.

AVDXBEP=9.2875 - AFLX....all coordinates and distances are measured in feet

AVDYBEP=0.0

AVDZBEP=-2.125

Where: AVDXBEP, AVDYBEP, AVDZBEP are eyepoint coordinates in Body Axis System

AFLX is aircraft X axis center of gravity (C.G.) position

9.2875 is C.G. to eyepoint position

0.0 is C.G. to eyepoint Y axis distance

-2.125 is C.G. to eyepoint Z axis distance

The eyepoint position relative to the Body Axis System is now converted into the CIG Earth Axis System using the body to earth transfer matrix derived above. These vector components represent the position offset between the C.G. and eyepoint in the Earth Axis System:

AVDXEEP=AVDXBEP*AVBDER11+AVDYBEP*AVBDER12+AVDZBEP*AVBDER13

AVDYEERP=AVDXBEP*AVBDER21+AVDYBEP*AVBDER22+AVDZBEP*AVBDER23

AVDZEERP=AVDXBEP*AVBDER31+AVDYBEP*AVBDER32+AVDZBEP*AVBDER33

Where: AVDXEEP, AVDYEERP, AVDZEERP are eyepoint vector components in the CIG Earth Axis System

AVDXBEP, AVDYBEP, AVDZBEP are eyepoint vector components in the Body Axis System

AVBDER11-33 are eyepoint to earth axis transfer matrix components

To get the eyepoint position in the CIG Earth Axis System, we simply added the eyepoint position offset to the C.G. position, since both positions are now defined in the CIG Earth Axis System.

AVPXCEP=ANAGEX + AVDXEEP.....all positions are measured in feet

AVPYCEP=ANAGEY + AVDYEERP

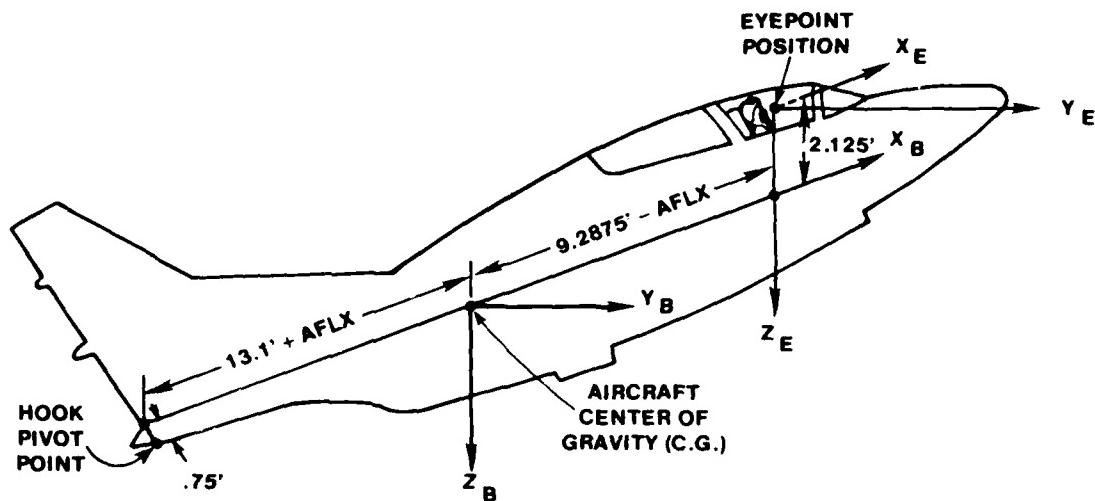
AVPZCEP=ANAGEZ + AVDZEERP

Where: AVPXCEP, AVPYCEP, AVPZCEP are eyepoint position, CIG Earth Axis System

ANAGEX, ANAGEY, ANAGEZ are aircraft C.G. position, CIG Earth Axis System

AVDXEEP, AVDYEERP, AVDZEERP are eyepoint position offsets from aircraft C.G., CIG Earth Axis System

NAVTRAEEQUIPCEN IH-335



NOTE: AIRCRAFT C.G. WILL SHIFT ALONG BODY AXIS AS AIRCRAFT FUEL, ORDNANCE WEIGHTS VARY, BUT EYEPOINT POSITION REMAINS FIXED RELATIVE TO AIRCRAFT FUSELAGE.

LEGEND:

- AFLX = VARIATION IN C.G. POSITION RELATIVE TO AIRCRAFT FUSELAGE DUE TO AIRCRAFT WEIGHT VARIATION
- C.G. = AIRCRAFT CENTER OF GRAVITY; POINT ABOUT WHICH AIRCRAFT BODY AXIS ROTATES
- EYEPOINT = POSITION OF PILOT'S EYEPOINT AND ORIGIN OF EYEPOINT AXIS SYSTEM
- HOOK PIVOT POINT = POSITION ON AIRCRAFT FUSELAGE ABOUT WHICH TAILHOOK PIVOTS (ONLY USED AS REFERENCE POINT, HERE)

Figure 9. Relationship Between Aircraft Body Axis and Eyepoint Axis Systems.

Since the Body and Eyepoint Axis Systems have their X, Y, and Z axis aligned (Figure 9), their attitude angles in the CIG Earth Axis System will be the same. These are derived below from the sine and cosine of yaw, pitch, and roll which are available from the flight software.

NAVTRAEEQUIPCEN IH-335

AVWVPYW=ATAN(SPSI/CPSI)
AVWVPPT=ATAN(AFSTHT/AFCTHT)
AVWVPRL=ATAN(AFSPHI/AFCPHI)

Where: AVWVPYW, AVWVPPT, AVWVPRL are CIG earth axis yaw, pitch, and roll in degrees
SPSI, CPSI are sine and cosine of yaw
AFSTHT, AFCTHT are sine and cosine of pitch
AFSPHI, AFCPHI are sine and cosine of roll

Next, extrapolations are computed for viewpoint position and attitude. These are required for the CIG System, which runs at a 60 Hertz rate, while visual runs at 30 Hertz. Hence, visual software must extrapolate ahead 16.7 milliseconds. A simple two-point forward extrapolation is used for both the viewpoint positions and angles.

Newton's Law is used as follows:

$$\begin{aligned} S' &= S + V*T/2 \quad (\text{predicted distance for } 1/2 \text{ field (16.7 ms)}) \\ S' &= S + (S-S_0)/2 \\ S' &= 1.5*S - .5*S_0 \end{aligned}$$

Where: S' is predicted distance traveled for 16.7 ms
 S_0 is previous position last frame (33 ms ago)
 V is average velocity between this frame and last frame
 S is position this frame
 T is $1/30$ (33 ms) = SEL frame (update period)
 $T/2$ is $1/60$ (16.7 ms) = CIG frame (update period)

The X, Y, Z position extrapolations are then computed:

$$\begin{aligned} AVELXEEP &= 1.5*AVPXCEP - .5*AVPXCEP1 \\ AVELYEEP &= 1.5*AVPYCEP - .5*AVPYCEP1 \\ AVELZEEP &= 1.5*AVPZCEP - .5*AVPZCEP1 \end{aligned}$$

Save previous frame's position:

$$\begin{aligned} AVPXCEP1 &= AVPXCEP \\ AVPYCEP1 &= AVPYCEP \\ AVPZCEP1 &= AVPZCEP \end{aligned}$$

Where: AVELXEEP, AVELYEEP, AVELZEEP are 16.7 ms position extrapolations
AVPXCEP, AVPYCEP, AVPZCEP are eyepoint position, CIG Earth axis, this frame values
AVPXCEP1, AVPYCEP1, AVPZCEP1 are eyepoint position, CIG Earth axis, last frame values

Similarly yaw, pitch, and roll angle extrapolations are computed.

$AVWVPRLD = 1.5 * AVWPRL - .5 * AVWPRL1$
 $AVWVPTD = 1.5 * AVWPPT - .5 * AVWPPT1$
 $AVWVPYWD = 1.5 * AVWPYW - .5 * AVWPYW1$

Save previous frame's attitude:

$AVWPRL1 = AVWPRL$
 $AVWPPT1 = AVWPPT$
 $AVWPYW1 = AVWPYW$

Where: $AVWVPRLD$, $AVWVPTD$, $AVWPWD$ are 16.7 ms attitude extrapolations
 $AVWPRL$, $AVWPPT$, $AVWPYW$ are eyepoint attitude, CIG Earth axis, this frame values
 $AVWPRL1$, $AVWPPT1$, $AVWPYW1$ are eyepoint attitude, CIG Earth axis, last frame values

Now the target parameters are processed. The CIG database has a total of 14 targets that can be displayed on the VTRS dome. There are a maximum of 6 moving targets (moving models) and 8 fixed targets that can be contained in a CIG database. However, typically, only 2 moving models are used as targets for any given database. Therefore, the flight software is only designed to handle 10 (2 moving and 8 fixed) targets at once. The positions of these 10 targets are passed from visual to flight to permit target locations to be displayed by the TACAN in the cockpit.

This module also computes the target to earth transfer matrix which is used by the Window/Point of Interest Module. The Target Axis System is shown in Figure 10. Note that this axis system is identical to the Body Axis System. Hence, the derivation is exactly the same as the body to earth transfer matrix.

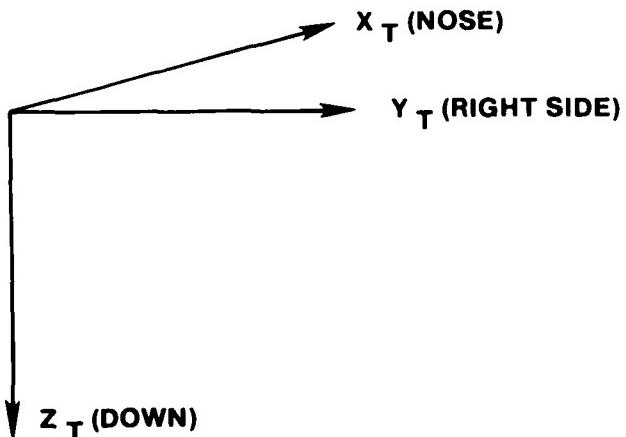


Figure 10. Target Axis System.

NAVTRAEOUIPCEN IH-335

First, attitude angles for the selected target (as explained above, 14 targets available) are used to compute sines and cosines required for the target to earth matrix derivation:

AVCMROLL=AVWMRL(IJPRTGI)
AVCMPITC=AVWMPT(IJPRTGI)
AVHEADCR=AVWMYW(IJPRTGI)

Compute sines and cosines for yaw, pitch, roll angles:

AVWSINR=SIN(AVCMROLL)
AVWCOSR=COS(AVCMROLL)
AVWSINP=SIN(AVCMPITC)
AVWCOSP=COS(AVCMPITC)
AVWSINY=SIN(AVHEADCR)
AVWCOSY=COS(AVHEADCR)

Where: AVCMROLL, AVCMPITC, AVHEADCR are target yaw, pitch, and roll in degrees
AVWMRL, AVWMPT, AVWMYW are selected target yaw, pitch, and roll angles in degrees
IJPRTGI is the target index (1-14)
AVWSINR, AVWCOSR are sine and cosine of roll
AVWSINP, AVWCOSP are sine and cosine of pitch
AVWSINY, AVWCOSY are sine and cosine of yaw

Next, the nine standard direction cosines that are used to convert from the Target Axis System (Figure 10) to the Flight Earth Axis System (Figure 6) are derived:

AVBEFL11=AVWCOSP*AVWCOSY
AVBEFL21=AVWCOSP*AVWSINY
AVBEFL31=-AVWSINP
AVBEFL12=AVWSINR*AVWSINP*AVWCOSY-AVWCOSR*AVWSINY
AVBEFL22=AVWCOSR*AVWCOSY+AVWSINR*AVWSINP*AVWSINY
AVBEFL32=AVWSINR*AVWCOSP
AVBEFL13=AVWSINR*AVWSINY+AVWCOSR*AVWSINP*AVWCOSY
AVBEFL23=AVWCOSR*AVWSINP*AVWSINY-AVWSINR*AVWCOSY
AVBEFL33=AVWCOSR*AVWCOSP

The Flight target to earth direction cosines are converted to CIG target to earth direction cosines.

AVBETS11=AVBEFL21
AVBETS21=AVBEFL11
AVBETS31=-AVBEFL31
AVBETS12=AVBEFL22
AVBETS22=AVBEFL12
AVBETS32=-AVBEFL32

NAVTRAEEQUIPCEN IH-335

AVBETS13=AVBEFL23
AVBETS23=AVBEFL13
AVBETS33==AVBEFL33

Where: AVBETS11-33 are target to earth transfer matrix components
in CIG System
AVBEFL11-33 are target to earth transfer matrix components
in Flight System

Flight logicals are passed to appropriate visual parameters. Separate visual parameters are used so that they may be manipulated independently of flight parameters. This feature protects flight logicals from being corrupted by changes in the visual software.

Finally, the last task is a test driver that allows the visual software and simulator to be controlled from the EOS. In this mode, all flight inputs are ignored and the operator must manually input viewpoint, target positions and attitudes. Flags can also be manually set. This is a convenient checkout mode which permits a controlled test, since it permits changing one parameter at a time (cause and effect are more easily identified).

SECTION III

WINDOW/POINT OF INTEREST MODULE

This module computes the vector from the observer to the selected target in Observer Axis System. This vector is called the point of interest (POI) vector. This vector is defined by a horizontal (azimuth) angle and vertical (elevation) angle. The sine and cosine of these angles are computed and passed to Visual/CIG Interface Module.

WINDOW/POINT OF INTEREST FUNCTIONAL DESCRIPTION.

This module defines "window" through which the observer must look in order to see the desired point of interest (POI) (Figure 11) in the Observer Axis System. The POI directs the observer's vision to the desired spot on the screen. It is usually the centroid of the target, but may be offset from the target centroid if desired. The angles defined in this routine are used by the CIG System to generate a target channel image and by the Target Image Projector Module to correctly point the target projector in the dome.

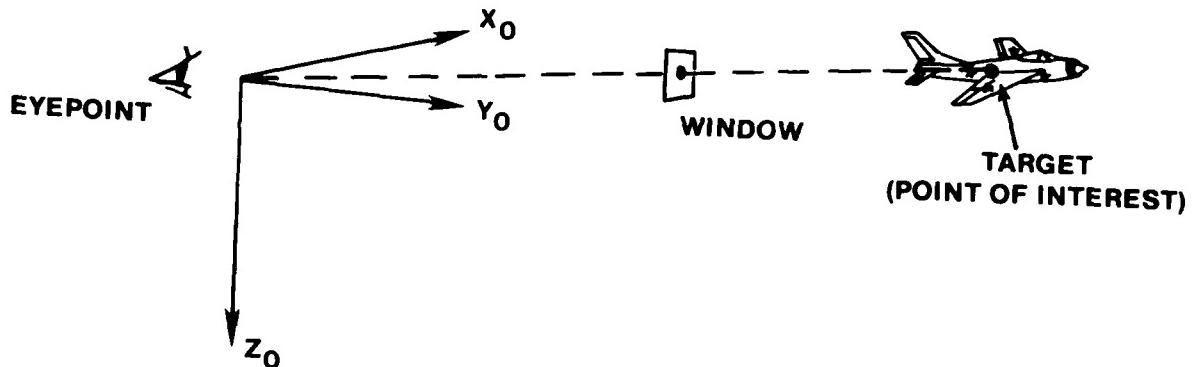


Figure 11. Point of Interest in Observer Axis.

The functional tasks of the Window/Point of Interest Module are as follows:

First, the POI for the selected target is determined. Each of the six moving targets has its own respective "offset" which is defined in its own Target Axis System. An example of the offset used in the carrier model is shown in Figure 12. Here, it is more desirable to aim the target projector at the carrier runway centerline rather than the carrier centroid, because it's extremely important to see the runway clearly when making carrier landings.

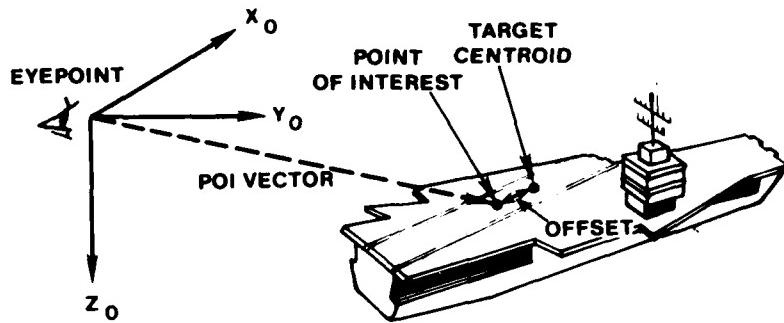


Figure 12. Point of Interest (POI) Offset in Observer Axis.

Next, after selecting the appropriate offset above, the components of this offset vector must be converted from the Target Axis System (in which they are defined) to the Earth Axis System.

Convert offset vector into earth axis vector components:

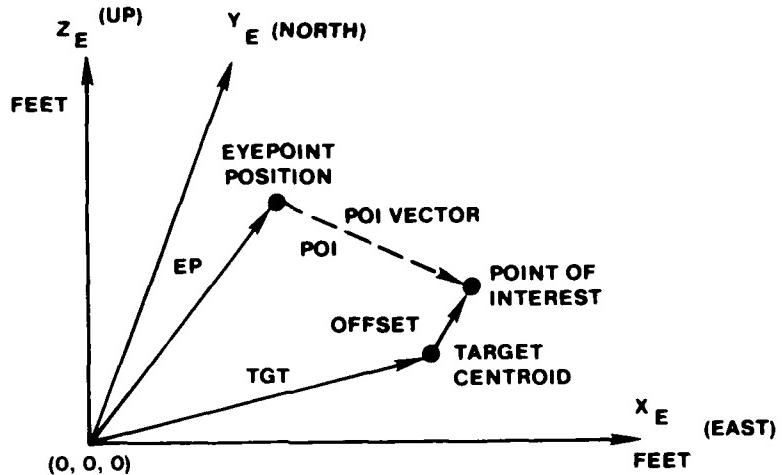
```

AVPXPOI=AVPXFLT1*AVBTES11+AVPYFLT1*AVBTES12+AVPZFLT1*AVBTES13
AVPYPOI=AVPXFLT1*AVBTES21+AVPYFLT1*AVBTES22+AVPZFLT1*AVBTES23
AVPZPOI=AVPXFLT1*AVBTES31+AVPZFLT1*AVBTES32+AVPZFLT1*AVBTES33

```

Where: AVPXPOI, AVPYPOI, AVPZPOI are offset vector components in the Earth Axis System, in units of feet
 AVPXFLT1, AVPYFLT1, AVPZFLT1 are offset vector components in the Target Axis System, in units of feet
 AVBTES11-AVBTES33 represents the body to earth transfer matrix

The vector from the observer to the point of interest (POI vector) is computed in the Earth Axis System (Figure 13). This vector is used to get the window angles and slant range (X, Y, Z distance) to the target. The POI vector is derived below:

**LEGEND:**

EP = EYEPOINT POSITION VECTOR (AVPXCEP, AVPYCEP, AVPZCEP)
POI = POINT OF INTEREST VECTOR (AVXPOIPG, AVYPOIPG, AVZPOIPG)
OFFSET = TARGET CENTROID TO POI VECTOR (AVPXPOI, AVPYPOI, AVPZPOI)
TGT = TARGET CENTROID POSITION VECTOR (AVPXPOIR, AVPYPOIR, AVPZPOIR)

Figure 13. Point of Interest in CIG Earth Axis System.

Define the sum of the total X, Y, Z vector components about a point for Figure 13.

$$\begin{aligned}
 \text{AVPXCEP} + \text{AVXPOIPG} - \text{AVPXPOI} - \text{AVPXPOIR} &= 0 \quad (\text{sum of vectors}=0) \\
 \text{AVPYCEP} + \text{AVYPOIPG} - \text{AVPYPOI} - \text{AVPYPOIR} &= 0 \quad (\text{sum of vectors}=0) \\
 \text{AVPZCEP} + \text{AVZPOIPG} - \text{AVPZPOI} - \text{AVPZPOIR} &= 0 \quad (\text{sum of vectors}=0)
 \end{aligned}$$

Where: AVPXCEP, AVPYCEP, AVPZCEP are eyepoint position vector components, Earth Axis in units of feet
 AVXPOIPG, AVYPOIPG, AVZPOIPG are eyepoint to POI vector components, Earth Axis in units of feet
 AVPXPOI, AVPYPOI, AVPZPOI are offset vector components in CIG Earth Axis in units of feet
 AVPXPOIR, AVPYPOIR, AVPZPOIR are target centroid position vector components, CIG Earth Axis in units of feet

NAVTRAEEQUIPCEN IH-335

Next solve for POI vector X, Y, Z components:

$$\begin{aligned} AVXPOIPG &= AVPXPOI + AVPXCEP \\ AVYPOIPG &= AVPYPOI + AVPYCEP \\ AVZPOIPG &= AVZPOI + AVZPOIR - AVZCEP \end{aligned}$$

...all vector components
are in units of feet

Now compute the slant range:

$$AVEORNGE = \sqrt{(AVXPOIPG^2 + AVYPOIPG^2 + AVZPOIPG^2)}$$

Where: AVEORNGE is eyepoint to point of interest vector magnitude
in feet

Then, the sine and cosine of the azimuth angle (in the X-Y plane) and elevation (in the XY-Z plane) of POI vector are computed. These angles completely define the orientation of the POI vector in the observer axis and are shown in Figure 14. The derivation is as follows:

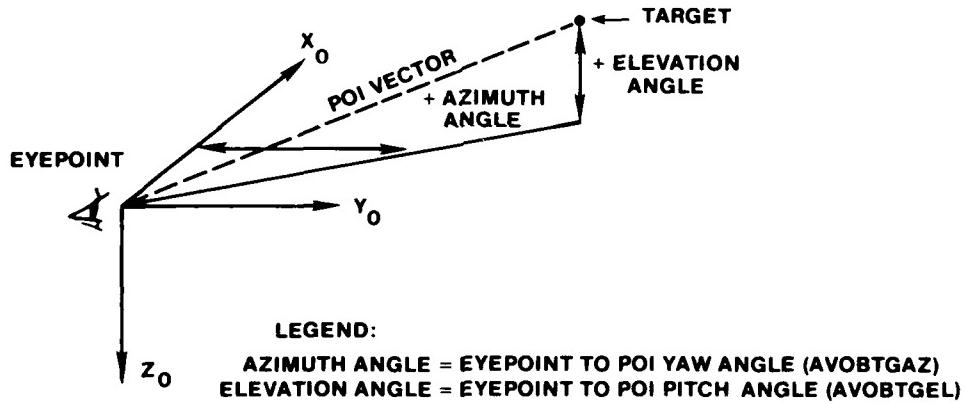


Figure 14. Point of Interest Angles in Observer Axis.

The POI unit vector is computed:

$$\begin{aligned} AVXPIPQN &= AVXPOIPG / AVEORNGE \\ AVYPIPQN &= AVYPOIPG / AVEORNGE \\ AVZPIPQN &= AVZPOIPG / AVEORNGE \end{aligned}$$

Where: AVXPIPQN, AVYPIPQN, AVZPIPQN are POI unit vectors, CIG Earth Axis System
 AVXPOIPG, AVYPOIPG, AVZPOIPG are eyepoint to POI vector components, CIG Earth Axis, in feet
 AVEORNGE is eyepoint to POI vector magnitude in feet

NAVTRAEEQUIPCEN IH-335

The POI unit vectors are now converted to the Eyepoint Axis System:

AVXPOIVO=AVXPIPZN*AVBDER11+AVYPIPZN*AVBDER21+AVZPIPZN*AVBDER31
AVYPOIVO=AVXPIPZN*AVBDER12+AVYPIPZN*AVBDER22+AVZPIPZN*AVBDER32
AVZPOIVO=AVXPIPZN*AVBDER13+AVYPIPZN*AVBDER23+AVZPIPZN*AVBDER33

Where: AVXPOIVO, AVYPOIVO, AVZPOIVO are POI unit vectors in the
Eyepoint Axis System
AVBDER11-AVBDER33 are eyepoint to earth axis transfer
matrix components
AVXPIPZN, AVYPIPZN, AVZPIPZN are POI unit vectors, CIG
Earth Axis System, in feet

Next, the sine and cosine of azimuth and pitch are computed in the
Eyepoint Axis System:

AVRPOIVP=SQRT(AVXPOIVO*AVXPOIVO+AVYPOIVO*AVYPOIVO)
AVSHDPOI=AVYPOIVO/AVRPOIVP
AVCHDPOI=AVXPOIVO/AVRPOIVP
AVSPHPOI=-AVZPOIVO
AVCPHPOI=AVRPOIVP

Where: AVRPOIVP is the length of the POI vector projected on
the X-Y plane
AVSHDPOI, AVCHDPOI are the sine and cosine of the POI
azimuth angle, Eyepoint Axis
AVSPHPOI, AVCPHPOI are the sine and cosine of the POI
elevation angle, Eyepoint Axis
AVXPOIVO, AVYPOIVO, AVZPOIVO are POI unit vectors in the
Eyepoint Axis System

Then, extrapolations for these sines and cosines are computed for the CIG
System. These are simple two-point extrapolations identical to the ones used
for the viewpoint parameters in Section II. Note that the angle rather than
the sines and cosines are extrapolated, because it was determined
experimentally that angles give a smoother and more consistent extrapolation.

First, the sines and cosines of azimuth and elevation are converted
into angles:

AVOBTGZ=ATAN(AVYPOIVO/AVXPOIVO)
AVOBTGEL=ATAN(-AVZPOIVO/AVRPOIVP)

Where: AVOBTGZ, AVOBTGEL are eyepoint to POI yaw and pitch
angles in degrees
AVXPOIVO, AVYPOIVO, AVZPOIVO are X, Y, Z vector
components of eyepoint to POI vector, eyepoint
axis system
AVRPOIVP is the length of the POI vector projected
on the X-Y plane

Next, a 16.7 millesecond forward extrapolation for the azimuth and elevation angles is computed for the CIG system:

$$\begin{aligned} \text{AVOBTAZ2} &= 1.5 * \text{AVOBTAZ1} - .5 * \text{AVOBTAZ1} \\ \text{AVOBTEL2} &= 1.5 * \text{AVOBTEL1} - .5 * \text{AVOBTEL1} \end{aligned}$$

Where: AVOBTAZ2, AVOBTEL2 are extrapolated eyepoint to POI yaw and pitch angles, eyepoint axis, in degrees
 AVOBTGZ, AVOBTGEL are current eyepoint to POI yaw and pitch angles, in degrees
 AVOBTAZ1, AVOBTEL1 are last frames yaw and pitch angles

Save the previous frame azimuth and elevation angles:

$$\begin{aligned} \text{AVOBTAZ1} &= \text{AVOBTAZ1} \\ \text{AVOBTEL1} &= \text{AVOBTEL1} \end{aligned}$$

Convert the azimuth and elevation values into sines and cosines:

$$\begin{aligned} \text{AVSHDPOB} &= \text{SIN}(\text{AVOBTAZ2}) \\ \text{AVCHDPOB} &= \text{COS}(\text{AVOBTAZ2}) \\ \text{AVSPHPOB} &= \text{SIN}(\text{AVOBTEL2}) \\ \text{AVCPHPOB} &= \text{COS}(\text{AVOBTEL2}) \end{aligned}$$

Where: AVSHDPOB, AVCHDPOB are the sine and cosine of the POI azimuth angle, field 2 value
 AVSPHPOB, AVCPHPOB are the sine and cosine of the POI elevation angle, field 2 value

Finally, a flag is tested to determine if the aircraft is on the ground (weight on wheels flag). If so, then the point of interest is set straight ahead. This feature provides the correct picture for carrier and airport landings.

The modules that interface this module and their parameters are:

Input Module:	(1) ZM6SW#33	- viewpoint position, CIG Earth Axis
		- selected POI position, CIG Earth Axis
		- target to earth transfer matrix
		- body to earth transfer matrix
	(2) ZMVISINT	- target alignment position, CIG Target axis system

NAVTRAEEQUIPCEN IH-335

Output Module: (1) ZM6SWCGI - sine, cosine of POI elevation and azimuth
 angles (in degrees), CIG Body axis
 system
 - sine, cosine of POI elevation and azimuth
 angles extrapolated for field 2
 updates, CIG Body axis system
(2) ZM6SW#37 - sine, cosine of POI elevation and azimuth
 angles, CIG Body axis system
 - POI to target vector magnitude (slant
 range) in feet

SECTION IV

TARGET IMAGE PROJECTOR

This module inputs azimuth and elevation angles from the Point of Interest Module and converts them into Target Image Projector (TIP) Axis System. These angles are then used to drive the TIP servos in the dome. This puts the target in the proper position in the dome. The target channel field of view is also controlled by this module through the use of a zoom ratio parameter which drives a zoom lens. This permits optimization of target resolution in the target channel.

TARGET IMAGE PROJECTOR MODULE FUNCTIONAL DESCRIPTION.

The TIP is physically offset 42.5 inches above and 6 inches behind the pilot's eyepoint in the dome (Figure 15). Also, the TIP pivots on a 6-inch arm to the right of the pilot's eyepoint. These offsets require that the eyepoint angles be converted into the TIP Axis System (Figure 16). These angles are translated into output commands that are sent to the servos to position the projector.

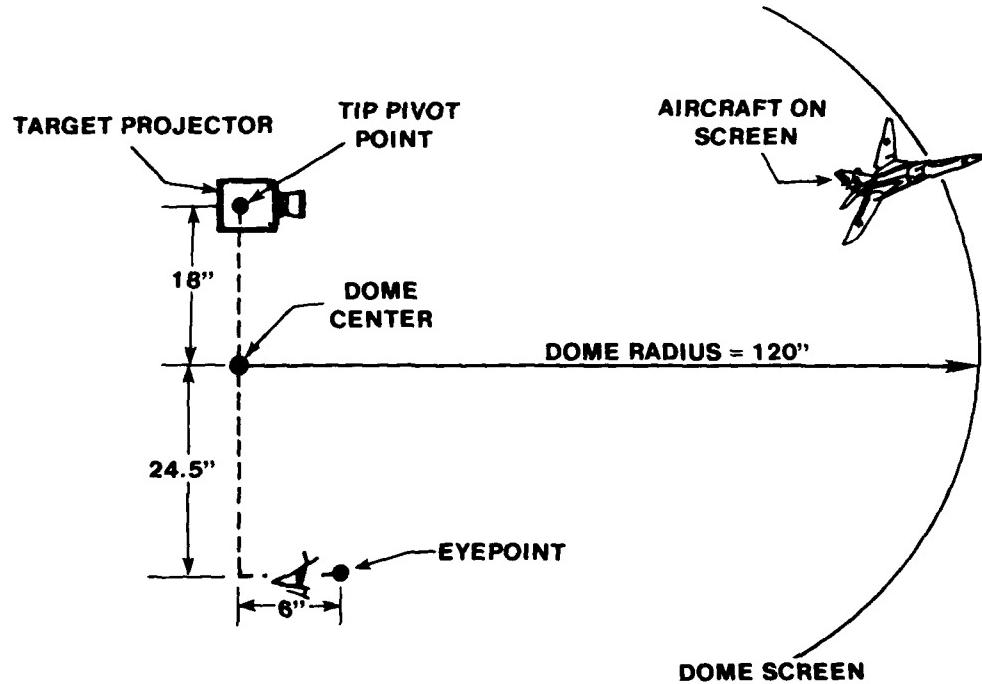


Figure 15. Target Image Projector Position Relative to Eyepoint, Side View.

NAVTRAEEQUIPCEN IH-335

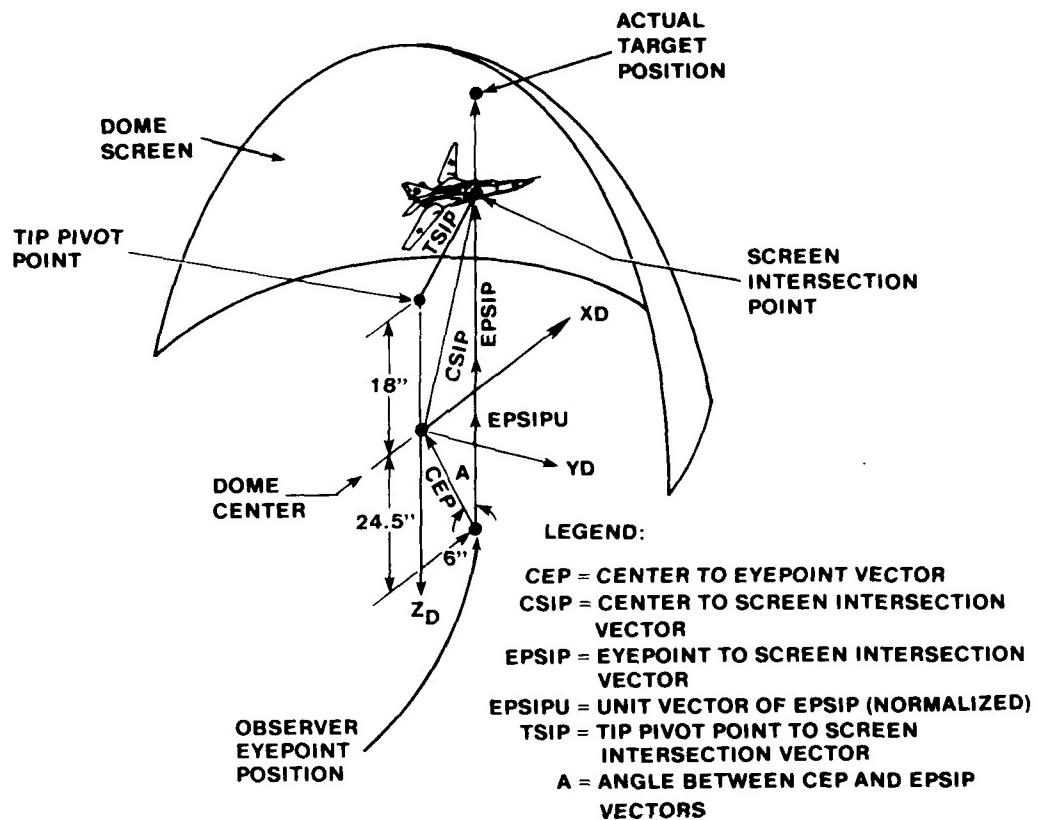


Figure 16. Screen Intersection Vectors in Display Axis System.

Referring to Figure 16, the screen intersection point is computed relative to the TIP pivot point using direction cosines, the dot product, the law of cosines, and the quadratic equation.

First the direction cosines of the POI vector in the Observer Axis System are computed. These direction cosines define the unit vector from the eyepoint to screen intersection point (EPSIPU):

$$\begin{aligned}
 \text{AVCHDPOI} &= X/\text{SQRT}(X*X + Y*Y) && \dots \text{adjacent over hypotenuse} \\
 \text{AVCPHPOI} &= \text{SQRT}(X*X+Y*Y)/\text{SQRT}(X*X+Y*Y+Z*Z) \\
 \text{AVSHDPOI} &= Y/\text{SQRT}(X*X + Y*Y) && \dots \text{opposite over hypotenuse} \\
 \text{AVSPHPOI} &= Z/\text{SQRT}(X*X+Y*Y+Z*Z)
 \end{aligned}$$

Where: X, Y, Z are the X, Y, Z components of the vector from eyepoint to screen intersection point in the Observer Axis System
 AVCHDPOI, AVSHDPOI are the cosine and sine of the POI azimuth angle in the Observer Axis System
 AVCPHPOI, AVSPHPOI are the cosine and sine of the POI elevation angle in the Observer Axis System

Let AVXPOIOB=X; AVYPOIOB=Y; AVZPOIOB=Z.

NAVTRAEEQUIPCEN IH-335

By substituting for X, Y, and Z, the direction cosines of the unit vector EPSIPU can be expressed:

$$\begin{aligned} \text{AVXPOIOB} &= \text{AVCHDPOI} * \text{AVCPHPOI} && (\text{X component of EPSIPU}) \\ \text{AVYPOIOB} &= \text{AVSHDPOI} * \text{AVCPHPOI} && (\text{Y component of EPSIPU}) \\ \text{AVZPOIOB} &= -\text{AVSPHPOI} && (\text{Z component of EPSIPU}) \end{aligned}$$

Where: AVXPOIOB, AVYPOIOB, AVZPOIOB are the direction cosines for the POI vector in the Observer Axis
AVCHDPOI, AVSHDPOI are the sine and cosine of the POI azimuth angle in the Observer Axis
AVCPHPOI, AVSPHPOI are the sine and cosine of the POI elevation angle in the Observer Axis

Next, the dot product between the dome center to observer vector (CEP) and the unit vector of observer to POI (EPSIPU) vector is computed (see Figure 16):

$$\begin{aligned} \text{AVCPC2OP} &= \text{CEP.EPSIPU} = \text{A1} * \text{A2} + \text{C1} * \text{C2} && (\text{dot product} = \text{sum of product of vector components}) \\ \text{AVCPC2OP} &= 6 * \text{AVXPOIOB} + 24.5 * \text{AVZPOIOB} && (\text{substitute vector components - vector in x-z plane only}) \end{aligned}$$

Where: AVCPC2OP is the DOT product of CEP and EPSIP unit vector
CEP is the dome center to observer vector
EPSIPU is the unit vector of EPSIP
A1, A2 are X components of CEP, EPSIPU vectors
C1, C2 are Z components of CEP, EPSIPU vectors
AVXPOIOB, AVYPOIOB are the direction cosines for the POI vector in the Observer Axis
6.0 is dome center to eyepoint distance along the dome X axis, in inches
24.5 is distance from dome center to eyepoint along the dome Z axis, in inches

Then, the distance from the observer to the intersection point on the screen (EPSIP) is computed using the law of cosines (L.C.) and the quadratic equation (Q.E.):

$$\begin{aligned} \text{CEP} * \text{CEP} &= \text{EPSIP} * \text{EPSIP} + \text{CSIP} * \text{CSIP} - 2 * \text{EPSIP} * \text{EPSIP} * \text{COSA} && (\text{define EPSIP using L.C.}) \\ 14400 &= \text{EPSIP} * \text{EPSIP} + 636.25 - 2 * \text{EPSIP} * \text{CSIP} * \text{COSA} && (\text{substitute dome parameters}) \\ \text{EPSIP} * \text{EPSIP} - 2 * \text{CSIP} * \text{COSA} * \text{EPSIP} - 13763.25 &= 0 && (\text{rearrange terms}) \\ \text{EPSIP} &= (2 * \text{CSIP} * \text{COSA} + \text{SQRT}((2 * \text{CSIP} * \text{COSA}) ** 2 - 4 * (-13763.75)) / 2 && (\text{solve for EPSIP using Q.E.}) \\ \text{EPSIP} &= \text{CSIP} * \text{COSA} + \text{SQRT}((\text{CSIP} * \text{COSA} + \text{CSIP} * \text{COSA}) + 13763.75) && (\text{arrange for substitution}) \end{aligned}$$

NAVTRAEEQUIPCEN IH-335

Substituting AVDSOBPI for EPSIP and AVCPC2OP for CSIP*COSA, the eyepoint to screen distance is determined:

$$AVDSOBPI = AVCPC2OP + \sqrt{AVCOC2OP * AVCOC2OP + 13763.75}$$

Where: $AVDSOBPI = EPSIP$ = eyepoint to screen intersection point distance
 $AVCPC2OP = CSIP * COSA$ = DOT product of the vector CEP and unit vector EPSIP
 $14400 = 120 * 120$ = dome radius distance squared (units in square inches)
 $636.25 = 6 * 6 + 24.5 * 24.5$ = CEP distance squared (units in square inches)
 $13763.75 = 14400 - 636.25$

Now the screen intersection point is defined in terms of the TIP Axis System (located at the TIP pivot point) X, Y, Z components:

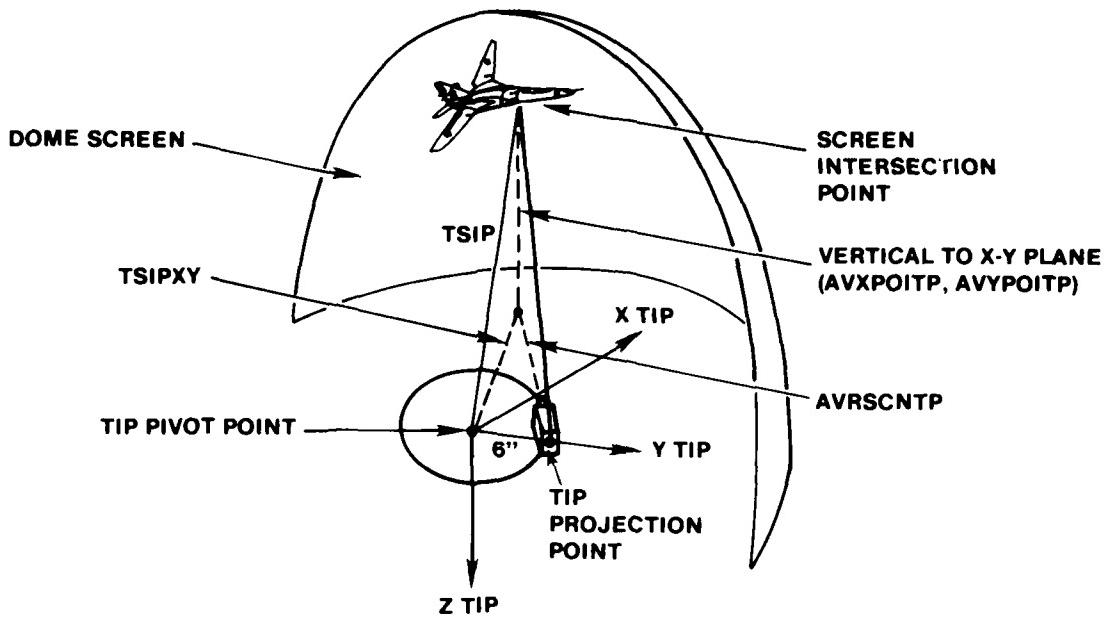
$$\begin{aligned} AVXPOITP &= 6. + AVDSOBPI * AVXPOIOB \\ AVYPOITP &= AVDSOBPI * AVYPOIOB \\ AVZPOITP &= 42.5 + AVDSOBPI * AVZPOIOB \end{aligned}$$

Where: $AVXPOITP$, $AVYPOITP$, $AVZPOITP$ are the screen intersection point coordinates in the TIP Axis System
6. = X component of TIP distance from the eyepoint (units in inches)
42.5 = the Z component of the TIP distance from the eyepoint (inches)
 $AVDSOBPI * AVXPOIOB$ is X observer to POI on screen distance
 $AVDSOBPI * AVYPOIOB$ is Y observer to POI on screen distance
 $AVDSOBPI * AVZPOIOB$ is Z observer to POI on screen distance

Next, the vector AVRSCNTP (Figure 17) is computed. This is used in deriving the TIP azimuth angle:

$$\begin{aligned} AVRSCNPI &= TSIPXY * TSIPXY \\ AVRSCNPI &= AVXPOITP * AVXPOITP + AVYPOITP * AVYPOITP \\ AVRSCNTP &= \sqrt{AVRSCNPI - 36.} \end{aligned}$$

Where: $AVRSCNPI$ is $TSIPXY$ squared
 $AVRSCNTP$ is TIP projection point to $(AVXPOITP, AVYPOITP)$
 $AVXPOITP, AVYPOITP$ X, Y components of screen intersection point projection on TIP X-Y plane
 $TSIPXY$ is projection of vector $TSIP$ on TIP X-Y plane
36. = $6 * 6$ = TIP pivot arm distance squared

**LEGEND:**

AVRSCNPI = PROJECTION OF THE VECTOR **TSIP** ON TIP X-Y PLANE
AVRSCNTP = TIP PROJECTION POINT TO THE POINT (**AVXPOITP, AVYPOITP**)
TSIP = TIP PIVOT POINT TO SCREEN INTERSECTION VECTOR
TSIPXY = PROJECTION OF THE VECTOR **TSIP** ON TIP X-Y PLANE

Figure 17. The Azimuth Vectors in the Tip Axis System.

Referring to Figure 18, the TIP azimuth angle in the TIP Axis System is derived using trigimetric relations as follows:

$$\begin{aligned}
 \text{TIP azimuth} &= H = A + B - 90. \\
 \sin(H) &= \sin(A+B-90.) = -\cos(A+B) = -\cos A \cos B + \sin A \sin B \\
 \cos(H) &= \cos(A+B-90.) = \sin(A+B) = \sin A \cos B + \cos A \sin B \\
 \sin(A) &= \text{AVYPOITP}/\sqrt{(\text{AVRSCNPI})} \\
 \cos(A) &= \text{AVXPOITP}/\sqrt{(\text{AVRSCNPI})} \\
 \sin(B) &= \text{AVRSCNTP}/\sqrt{(\text{AVRSCNPT})} \\
 \cos(B) &= 6./\sqrt{(\text{AVRSCNPI})} \\
 \sin(H) &= \text{AVYPOITP}*6./\text{AVRSCNPI} + \text{AVXPOITP}*\text{AVRSCNTP}/\text{AVRSCNPI} \\
 \cos(H) &= \text{AVYPOITP}*6./\text{AVRSCNPI} - \text{AVXPOITP}*\text{AVRSCNTP}/\text{AVRSCNPI} \\
 \cos(H) &= \text{AVCTPHDG} = \text{AVRSCNTP}*\text{AVXPOITP} + \text{AVYPOITP}*6. \\
 \sin(H) &= \text{AVSTPHDG} = \text{AVRSCNTP}*\text{AVYPOITP} - \text{AVXPOITP}*6. \\
 H &= \text{AVTIPHGDG} = \text{ATAN}(\text{AVSTPHDG}/\text{AVCTPHDG})
 \end{aligned}$$

Where:

- A = angle between TIP X axis and vector TSIPXY
- B = angle between vector AVRSCNPI and TIP pivot arm
- H = TIP azimuth angle
- AVRSCNPI = TSIPXY squared
- AVRSCNTP = TIP projection point to (AVXPOITP, AVYPOITP)
- AVXPOITP, AVYPOITP are X, Y components of vector TSIPXY
- 6. = TIP pivot arm length (inches)

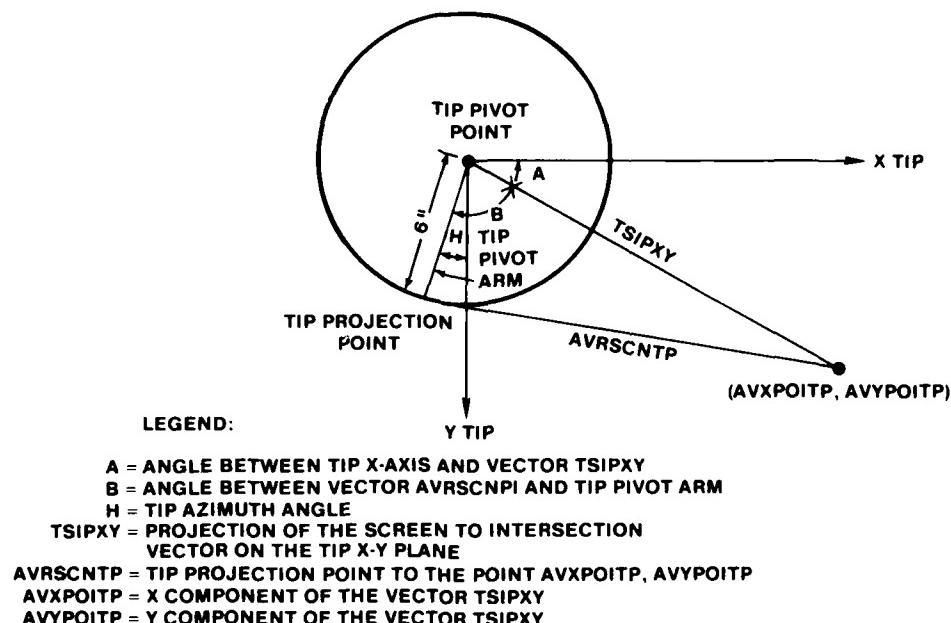


Figure 18. Tip Azimuth Angles in Tip Axis System, Top View.

The TIP evaluation angle in the TIP Axis System is easily computed:

$$\text{AVTIPPTH} = \text{ATAN}(-\text{AVZPOITP}/\text{AVRSCNTP})$$

Where:

- AVTIPPTH is the TIP elevation angle in degrees
- AVZPOITP is the Z component of vector TSIP
- AVRSCNTP is TIP projection point to (AVXPOITP, AVYPOITP)

The TIP zoom ratio is computed as a function of the slant range from the zoom schedule in "WTIPZOOM."

The TI^D roll is computed from TIP heading and pitch commands. (It removes roll induced by TIP azimuth and elevation.)

NAVTRAEEQUIPCEN IH-335

TIP position is checked against excursion limits and drive commands limited and iris closed to protect hardware from hitting stops in dome.

a. "TIPSAME" determines if TIP was in stop area last frame.

(1) Used to determine if iris should stay close.

(2) Used in setting proper commanded headings based on current and previous commanded headings.

TIP elevation and azimuth commands are limited to 200 deg/sec by "LIMITEL" and "LIMITAZ" commands.

a. TIP velocity and acceleration for AZ and EL are computed and then commanded positions are limited for 200 deg/sec.

(1) TIP, azimuth, elevation, roll commands are delayed one frame (33 millisec) to sync TIP hardware with CIG background video.

(2) TIP, azimuth, elevation, roll, iris commands are scaled to drive hardware.

Extrapolations (field 2 updates) for TIP elevation, azimuth, roll angles are computed for the CIG System. A simple two-point extrapolation (see Section II) is used.

The extrapolated yaw, pitch, and roll angles are computed as follows:

$$\begin{aligned} \text{AVWTRL2} &= 1.5 * \text{AVTIPROL} - .5 * \text{AVWTRL11} \\ \text{AVWTPT2} &= 1.5 * \text{AVTIPPTH} - .5 * \text{AVWTPT11} \\ \text{AVWTYW2} &= 1.5 * \text{AVTIPHDG} - .5 * \text{AVWTYW11} \end{aligned}$$

Where: AVWTYW2, AVWTPT2, AVWTRL2 are extrapolated TIP yaw, pitch, roll angles in TIP axis system
AVTIPHDG, AVTIPPTH, AVTIPROL are current TIP yaw, pitch, roll angles in TIP axis system
AVWTYW11, AVWTPT11, AVWTRL11 are last frame's TIP yaw, pitch, roll angles in TIP axis system

Save last frame's TIP yaw, pitch, roll angles:

$$\begin{aligned} \text{AVWTRL11} &= \text{AVTIPROL} \\ \text{AVWTPT11} &= \text{AVTIPPTH} \\ \text{AVWTYW11} &= \text{AVTIPHDG} \end{aligned}$$

The modules that interface this routine and their parameters are:

NAVTRAEEQUIPCEN IH-335

Input Module: ZM6SW#36 - sine, cosine of elevation, azimuth, POI
 angles, and CIG Body
WTIPZOOM - TIP zoom ratio (ratio varies from 1 to 0.1 to
 control target channel field of view)
LIMITAZ - commanded TIP, azimuth angle, TIP Axis
LIMITEL - commanded TIP elevation angle, TIP Axis
TIPSAME - TIP same side (T/F)

Output Module: ZM6SWCGI - elevation, azimuth, roll, zoom, TIP Axis
 - field 2 updates, TIP Axis

SECTION V

MOVING MODEL CONTROL

This module looks at the database loaded and then activates the appropriate subroutines to drive the models that exist in that environment. Each database has its own unique models that must be driven by the weapons visual software. The user may utilize a variety of moving models for each database, but must know which models are available for each respective database. A default mode exists for each database so that the unknowledgeable user may perform limited operations with the system.

MOVING MODEL CONTROL MODULE FUNCTIONAL DESCRIPTION.

This module is the "brains" of the visual weapon system and must decide what subroutines are permitted to run. This decision is based on which database is loaded and which user request flags are set. These two inputs are explained below:

First, the environment is checked to determine which options (Table 1) the user is permitted to exercise. For the DAYFORRESTAL and FLOLS Meridian, there are no options. The drivers for these two databases come up automatically. The other databases, however, have the options listed in Table 1. In the default mode, the tank and truck drivers are activated automatically for the Gunnery Range, Twin Towns, and River Valley. The six basic environment types are listed below.

- a. DAYFORRESTAL (aircraft carrier)
- b. Standard Meridian (has TA4J for formation flight)
- c. FLOLS Meridian (has carrier FLOLS on runway)
- d. Gunnery range (has tank, truck, rockets, bombs, bullets)
- e. Twin towns (has tank, truck, SAM, FLAK, rockets, bombs, bullets)
- f. River valley (has tank, truck, FLAK, rockets, bombs, bullets)

NAVTRAEEQUIPCEN IH-335

TABLE 1. MOVING MODEL OPTIONS FOR DATABASE ENVIRONMENTS

DATABASES:	MOVING MODEL DRIVERS								
	CVA	FLOLS	TA4J	ASM	TANK	TRUCK	SAM	FLAK	GEN
DAYFORRESTAL	X	X							X
STANDARD MERIDIAN			X						X
FLOLS MERIDIAN		X							X
GUNNERY RANGE				X	X	X			X
TWIN TOWNS				X	X	X	X	X	X
RIVER VALLEY				X	X	X		X	X

LEGEND:

CVA = US FORRESTAL AIRCRAFT CARRIER
 FLOLS = FRESNAL LENS OPTICAL LANDING SYSTEM
 TA4J = TA4J SKYHAWK AIRCRAFT
 ASM = AIR-TO-SURFACE MISSILE (2.75 INCH ROCKET)
 TANK = RUSSIAN TANK
 TRUCK = FIVE TRUCK CONVOY
 SAM = SURFACE-TO-AIR MISSILE (SA-3, SA-6)
 FLAK = ANTI-AIRCRAFT FLAK
 GEN = GENERAL PURPOSE MOVING MODEL DRIVER

NOTE: VARIATIONS FOR EACH OF THESE DATABASES ARE AVAILABLE

Valid user requests for moving model drivers results in the activation of one or more of the nine weapon subroutine drivers illustrated in Table 1. For example, the user may desire to use only the tank driver in Twin Towns even though six drivers are available. To perform this task the user must set the tank flag in Table 2 to true and set the other applicable flags (refer to twin towns in Table 1) to false. The nine subroutine drivers and the conditions necessary to activate them are summarized below:

TABLE 2. MOVING MODEL CONTROL FLAGS

CVA ¹	NA
FLOLS ²	NA
TA4J	EAFORFLT
ASM	EJBROB
TANK	LVTANK
TRUCK	LVTRUCK
SAM	EJBSAM
FLAK	EJBFLAK
GEN	EGENERAL

1 NOTE: AUTOMATICALLY ACTIVATED WHEN DAYFORRESTAL IS LOADED

2 NOTE: AUTOMATICALLY ACTIVATED WHEN FLOLS MERIDIAN IS LOADED

NAVTRAEEQUIPCEN IH-335

a. WCARRIER drives a model of the USS FORRESTAL aircraft carrier. This subroutine is automatically activated whenever the DAYFORRESTAL database is loaded.

b. WFLOLS drives a model of the Fresnal Lens Optical Landing System (FLOLS). This subroutine is automatically activated whenever the DAYFORRESTAL or FLOLS Meridian database is loaded.

c. WTA4J drives a model of the TA4J "SKYHAWK" aircraft. This subroutine is activated whenever the formation flight flag is set and standard Meridian database is loaded.

d. WROCKETS drives models of two Air-To-Surface (ASM) 2.75 inch rockets. This subroutine is activated whenever the ASM flag is set and Gunnery Range, Twin Towns, or River Valley databases are loaded.

e. WSAM drives a model of a Russian Surface-To-Air (SAM) missile (SA-3 or SA-6). This subroutine is activated whenever the SAM launch flag is set and the Twin Towns database is loaded.

f. WFLAK drives a model of anti-aircraft FLAK. This subroutine is activated whenever the FLAK flag is set and Twin Towns or River Valley databases are loaded.

g. WTANK drives a model of a Russian tank on each respective road system for Gunnery Range, Twin Towns, and River Valley. This routine is activated whenever the tank flag is set and one of the above three databases is loaded.

h. WTRUCK drives a model of a convoy of five trucks on each respective road system for Gunnery Range, Twin Towns, and River Valley. The trucks are driven on different roads than the tanks. This subroutine is activated whenever the truck flag is set and one of the above three databases is loaded.

i. WGENERAL is a general purpose driver available to the user for special applications when one of the above drivers fails to meet his requirements. This routine is activated whenever the general purpose flag is set and is independent of the database load.

The parameter relationship between input modules and the output module that ZM6SMOVE drives are as follows:

Input Module: ZM6SWCGI - active environment set (tells what database is loaded)
ZM6SW#33 - load guns, bombs, rockets, flags set

NAVTRAEEQUIPCEN IH-335

Output Module: WCARRIER - is called if carrier database loaded
WFLAK - called if FLAK launched and weapon*
database is loaded
WFLOLS - called if carrier or FLOLS Meridian
database is loaded
WGENERAL - is called if EGENERAL = true
WROCKETS - called if ASM launched and weapon
database is loaded
WSAM - called if SAM launched and twin town
database is loaded
WTA4J - called if EAFORFLT true and standard
Meridian loaded
WTANK - called if weapon database loaded and
LVTANK = true
WTRUCK - called if weapon database loaded and
LVTRUCK = true

* Note: Weapon databases include Gunnery Range, Twin Towns, and River Valley

SECTION VI

WEAPON SCORING

This module scores weapon deliveries of bombs, bullets, and rockets shot from the T2-C aircraft. Bombs are scored as ground miss distances. Rockets and bullets are scored as closest point of approach (CPA) above the ground. Weapon visual effect flags are also set by this module. These include hit flags for the tank and trucks, and impact flags for bombs, rockets, and bullets.

WEAPON SCORING MODULE FUNCTIONAL DESCRIPTION.

This module scores bombs by computing the ground impact distance from the selected target. This is illustrated in Figure 19. Note that all scoring vector magnitudes are measured in feet. The following is a summary of this derivation:

As noted in Figure 19, the following parameters are passed to this module:

- a. Weapons release vector (PA)
- b. Target vector (PT)
- c. Weapon impact vector (B')

The ground miss distance (Q), then is readily computed.

Express vector relationship of B' , Q , PT:

$$B' - Q - PT = 0 \quad \dots \text{sum of vectors around a point} = 0$$

Solve for target to impact vector Q :

$$\begin{aligned} Q &= B' - PT && \dots \text{solve for } Q \\ AJTXIMP &= AJPXIMP - AJTXPOS && \dots \text{define } Q, B', PT \text{ in X, Y, Z} \\ AJTYIMP &= AJPYIMP - AJTYPPOS && \text{vector components} \\ AJTZIMP &= AJPZIMP - AJTZPOS \end{aligned}$$

Where: AJTXIMP, AJTYIMP, AJTZIMP are target to impact vector
 (Q) components, CIG Earth Axis
AJPXIMP, AJPYIMP, AJPZIMP are weapon impact vector
 (B') components, CIG Earth Axis
AJTXPOS, AJTYPPOS, AJTZPOS are target vector (PT)
components, CIG Earth Axis

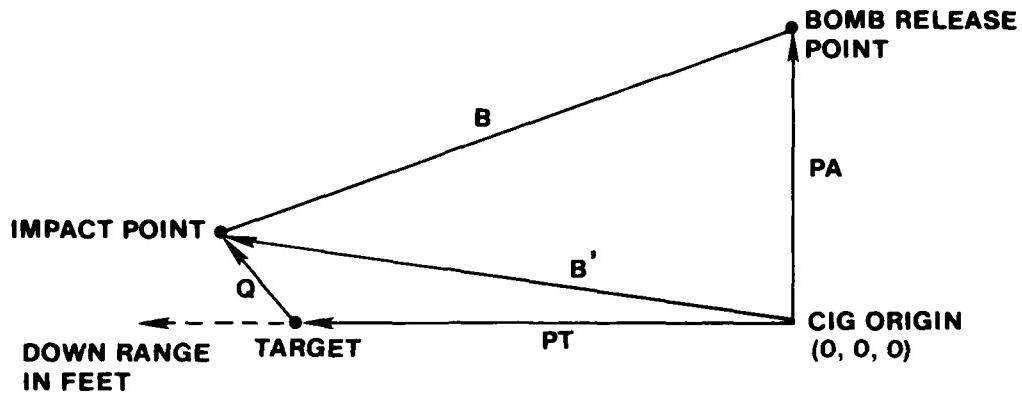


Figure 19. Vector Diagram for Bomb Miss Distance.

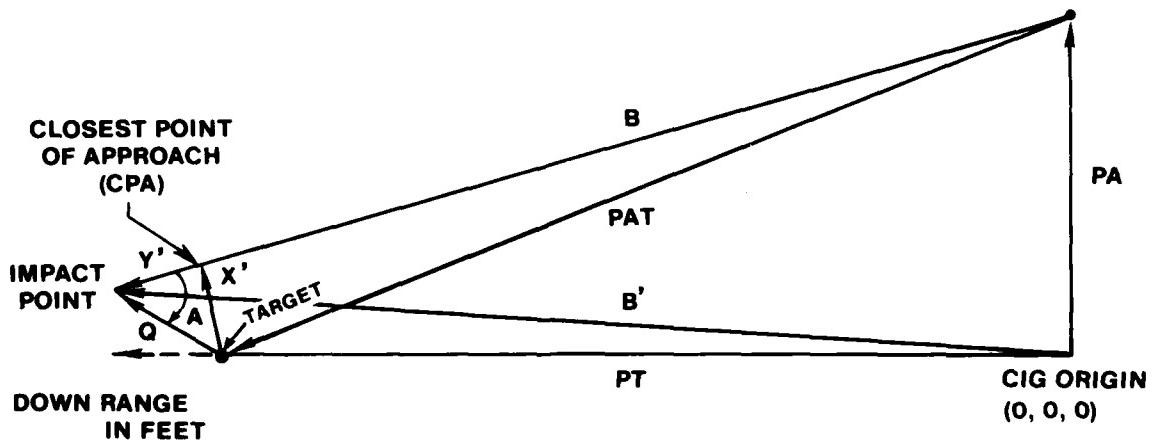
Solve for magnitude of vector Q:

$$AJTLEGQ2 = \text{SQRT}(AJTXIMP*AJTXIMP+AJTYIMP*AJTYIMP+AJTZIMP*AJTZIMP)$$

Where: $AJTLEGQ2 = Q$ = magnitude of target to impact vector

This module scores rockets and bullets by computing the closest point of approach (CPA) above ground. As shown in Figure 20, the CPA is the closest point the trajectory passes by the target. This is especially important for rockets and bullets, which tend to impact at low elevation angles (10-30 degrees). Hence, it is possible for a bullet to pass through a target, yet impact the ground hundreds of feet from the target.

ROCKET/BULLET RELEASE POINT



NOTE: FOR ROCKETS AND BULLETS, THE MISS DISTANCE IS DEFINED AS THE CLOSEST POINT OF APPROACH ABOVE THE GROUND (SEE TEXT)

Figure 20. Vector Diagram for Rocket/Bullet Miss Distance.

NAVTRAEEQUIPCEN IH-335

Using the vectors PA, PT, B', PAT, Q, AND Y', the closest point of approach vector X' can be derived as follows.

Express vector relationship of PA, PAT, PT:

$$PA + PAT - PT = 0 \quad \dots \text{sum of vectors around point} = 0$$

Solve for weapon to target vector PAT:

$$\begin{aligned} PAT &= PT - PA && \dots \text{solve for PAT} \\ AJATXPOS &= AJTXPOS - AJAXPOS && \dots \text{define PAT, PT, PA in X, Y, Z} \\ AJATYPOS &= AJTYPOS - AJAYPOS && \text{vector components} \\ AJATZPOS &= AJTZPOS - AJAZPOS \end{aligned}$$

Where: AJATXPOS, AJATYPOS, AJATZPOS are weapon (aircraft) to target X, Y, Z vector components, CIG Earth Axis
AJTXPOS, AJTYPOS, AJTZPOS are target X, Y, Z position vector components, CIG Earth Axis
AJAXPOS, AJAYPOS, AJAZPOS are weapon (aircraft) X, Y, Z position vector components, CIG Earth Axis system

Solve for magnitude of PAT:

$$AJATRPOS = \sqrt{(AJATXPOS^2 + AJATYPOS^2 + AJATZPOS^2)}$$

Where: AJATRPOS=PAT=magnitude of weapon to target vector

Express vector relationship of PA, B, B':

$$PA + B - B' = 0$$

Solve for weapon release point to impact vector B:

$$\begin{aligned} B &= B' - PA && \dots \text{solve for B} \\ AJAXIMP &= AJPXIMP - AJAXPOS && \dots \text{define B, B', PA in X, Y, Z} \\ AJAYIMP &= AJPYIMP - AJAYPOS && \text{vector components} \\ AJAZIMP &= AJPZIMP - AJAZPOS \end{aligned}$$

Where: AJAXIMP, AJAYIMP, AJAZIMP are X, Y, Z components of the weapon to impact vector
AJPXIMP, AJPYIMP, AJPZIMP are X, Y, Z components of impact position, CIG Earth Axis
AJAXPOS, AJAYPOS, AJAZPOS are X, Y, Z components of aircraft position, CIG Earth Axis

Solve for magnitude of vector B:

$$AJTLEGQ1 = \sqrt{(AJAXIMP^2 + AJAYIMP^2 + AJAZIMP^2)}$$

NAVTRAEEQUIPCEN IH-335

Where: $AJTLEGQ1=B$ =magnitude of weapon to impact vector

Solve for $\cos(A)$ in Figure 20, using the DOT product relationship:

$$\begin{aligned} Q \cdot B &= [\text{ABS}(Q) * \text{ABS}(B)] * \text{COSA} && \dots \text{DOT product definition} \\ \text{AJTLEGY} &= [\text{AJTLEGQ2} * \text{AJTLEGQ1}] * \text{AJCOSYQ} && \dots \text{substitute VIS PAR's.} \\ \text{AJTLEGY} &= [\text{AJTXIMP} * \text{AJAXIMP} + \\ &\quad \text{AJTYIMP} * \text{AJAYIMP} + \\ &\quad \text{AJTZIMP} * \text{AJAZIMP}] && \dots (\text{can be redefined in} \\ &&& X, Y, Z \text{ components}) \\ \text{AJCOSYQ} &= \text{AJCOSYQ}/(\text{AJTLEGQ2} * \text{AJTLEGQ1}) && \dots (\text{substituting and} \\ &&& \text{solving for AJCOSYQ}) \end{aligned}$$

Where: $Q \cdot B$ is DOT product of vectors Q and B
 $\text{ABS}(Q) * \text{ABS}(B)$ is product of absolute values of Q and B
 COSA is cosine of angle A (see Figure 20)
 $\text{AJTLEGY} = Q \cdot B$ =visual parameter for Q.B
 $\text{AJTLEGQ2}, \text{AJTLEGQ1}$ are vector magnitudes of Q and B
 $\text{AJCOSYQ} = \text{COSA}$ =visual parameter for COSA
 $\text{AJTXIMP}, \text{AJTYIMP}, \text{AJTZIMP}$ are target to impact vector
components, CIG Earth Axis
 $\text{AJAXIMP}, \text{AJAYIMP}, \text{AJAZIMP}$ are weapon to impact vector
components, CIG Earth Axis

Compute the unit vector components of B (Figure 20):

$$\begin{aligned} \text{AJUAXIMP} &= \text{AJAXIMP}/\text{AJTLEGQ1} && \dots \text{divide X, Y, Z components of B} \\ \text{AJUAYIMP} &= \text{AJAYIMP}/\text{AJTLEGQ1} && \dots \text{by absolute value of B} \\ \text{AJUAZIMP} &= \text{AJAZIMP}/\text{AJTLEGQ1} \end{aligned}$$

Where: $\text{AJUAXIMP}, \text{AJUAYIMP}, \text{AJUAZIMP}$ are unit vector components
of vector B
 $\text{AJAXIMP}, \text{AJAYIMP}, \text{AJAZIMP}$ are weapon to impact vector
components, CIG Earth Axis
 AJTLEGQ1 is the magnitude of vector B

Now, the vector Y' (see Figure 20) can be derived:

$$\begin{aligned} Y' &= Q * \text{COSA} && \dots \text{solving for } Y' \\ \text{AJYLEG} &= \text{AJTLEGQ2} * \text{AJCOSYQ} && \dots \text{substituting for VIS PAR's} \end{aligned}$$

Solving for X, Y, Z components of Y' :

$$\begin{aligned} \text{AJYLEGX} &= \text{AJYLEG} * \text{AJUAXIMP} && \dots \text{since vectors B and } Y' \text{ have} \\ \text{AJYLEGY} &= \text{AJYLEG} * \text{AJUAYIMP} && \text{same orientation, the unit} \\ \text{AJYLEGZ} &= \text{AJYLEG} * \text{AJUAZIMP} && \text{vectors of B define the unit} \\ &&& \text{vectors of } Y' \end{aligned}$$

NAVTRAEEQUIPCEN IH-335

Where: AJYLEGX, AJYLEGY, AJYLEGZ are vector components of Y'
 AJYLEG is the magnitude of vector Y'
 AJUAXIMP, AJUAYIMP, AJUAZIMP are unit vectors of B
 ...direction of Y'

Finally, the closest point of approach vector X' (Figure 20) can be computed:

$$\begin{aligned} Q - Y' - X' &= 0 && \dots \text{sum of vectors about point } = 0 \\ X' &= Q - Y' && \dots \text{solve for } X' \text{ (CPA)} \\ \text{AJXCPA} &= \text{AJTXIMP} - \text{AJYLEGX} && \dots \text{substitute VIS PAR's} \\ \text{AJYCPA} &= \text{AJTYIMP} - \text{AJYLEGY} \\ \text{AJZCPA} &= \text{AJTZIMP} - \text{AJZLEGZ} \end{aligned}$$

Where: AJXCPA, AJYCPA, AJZCPA are vector components of X',
 CIG Earth Axis System
 AJTXIMP, AJTYIMP, AJTZIMP are vector components of Q,
 CIG Earth Axis System
 AJYLEGX, AJYLEGY, AJYLEGZ are vector components of Y',
 CIG Earth Axis System

Solve for magnitude of X' which is the closest point of approach (CPA) that bullet/rocket passes by the target:

$$\text{AJCPA} = \text{SQRT}(\text{AJXCPA}^2 + \text{AJYCPA}^2 + \text{AJZCPA}^2) \quad \begin{aligned} &\dots \text{sum of vectors } X, Y, Z \\ &\dots \text{components} \end{aligned}$$

Where: AJCPA is the magnitude of the CPA vector X'
 AJXCPA, AJYCPA, AJZCPA are the vector components of X',
 CIG Earth Axis System

Next, since bullets and rockets do not significantly bore holes below the ground, the Z (vertical) component of the CPA vector must be checked for negative values. If the Z component is negative, then the ground miss distance Q is used instead of the CPA for scoring. Figure 21 illustrates this point. The mathematical derivation is provided below:

$$X'(Z) = AJZCPA$$

If AJZCPA < 0, then CPA miss distance is invalid and the value of Q is used.

NAVTRAEEQUIPCEN IH-335

Otherwise, CPA miss distance is valid so the value of X' can be used.

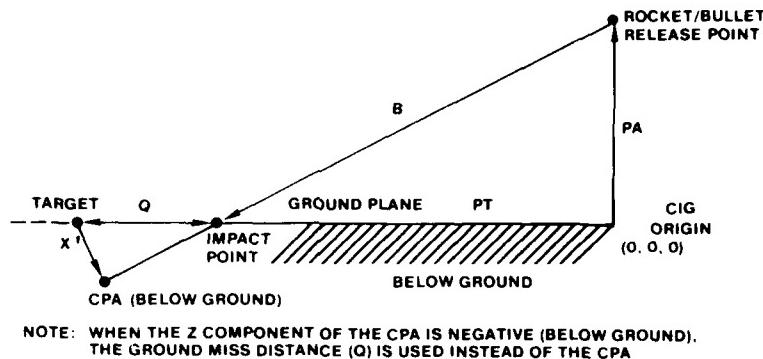


Figure 21. Vector Diagram for CPA Values Below the Ground.

Target hits are scored when the kill radius of the target and the weapon kill radius overlap. The kill radius of fixed targets are specified by the CIG System when a database is loaded. The kill radius of moving models are specified by the weapon visual software. Typically, moving models such as tanks and trucks have a small kill radius (30 feet), while fixed targets have a larger kill radius (100 feet, etc.). The kill radius of the weapons is listed below:

Bullets: 0.0 feet (must hit target directly).
 Rockets: 10.0 feet
 Bombs: 50-300 feet

This module also sets weapon control flags. These flags control the visual weapon effects and also synchronize weapon trajectories with the flight software. These flags are summarized:

Tank Explode Flag	...IVWMHT set to 3
Truck Explode Flag	...IVWMHT set to 4
Fixed Target Delete Flag	...IVWFHT set to 1, 2, 3, 4, 5, 6, 7, 8 to delete respective target
Bullet Impact Flag	...KVWGIF set to 2 for 2 bullet hits
Bomb Impact Flag	...KVWGIF set to 2 for 2 bomb hits
Rockets Impact Flag	...LDWRLF set to true for explode
Scoring Complete Flag	...EJSCOREF set to true when done
Target Killed	...EJTGDEAD set true when target hit

SECTION VII
VISUAL/CIG INTERFACE

This module interfaces weapon visual parameters and CIG parameters on the CIG side (Figure 1). All weapon visual parameters required to drive the CIG System are converted into PDP-11/55T format and put into datapool for output to the PDP. All CIG parameters required for weapon visual software are obtained from datapool and assigned to the appropriate weapon visual parameters. These are primarily database parameters and are required to control the air-to-ground software.

VISUAL/CIG INTERFACE MODULE FUNCTIONAL DESCRIPTION.

All interface tasks done by this module are performed on parameters that reside in datapool. The actual data transfer between the DEC and SEL Systems is performed by another subroutine called WDECSEL. This subroutine merely takes these datapool parameters which have been set up by the Visual/CIG Interface Module and reads or writes to the CIG System. The parameters in Tables 3 and 4 are real-time parameters and are transferred at a 30 Hertz rate. The parameters in Table 5 are database initialization parameters and are only transferred when a new database is loaded. These tables will be discussed below.

There are three basic interface tasks performed by this module:

a. The first is the transfer of all weapon visual parameters required to drive the CIG Visual System. Note that all position and attitude coordinates are in CIG Earth Axis system (Figure 7) and are specified in feet and degrees respectively. These parameters are shown in Table 3 and are summarized below:

- (1) Viewpoint position field 1 and field 2 (X, Y, Z positions)
- (2) Viewpoint attitude field 1 and field 2 (yaw, pitch, roll)
- (3) Moving models 1-6 position field 1 and field 2
- (4) Moving models 1-6 attitude field 1 and field 2
- (5) FLOLS glide slope, roll angle, brightness, meatball position
- (6) TIP roll, pitch, yaw angles field 1 and field 2
- (7) TIP zoom ratio (values: .15-1.0)
- (8) Window attitude: sine, cosine of yaw, pitch angles for field 1 and field 2

NAVTRAEEQUIPCEN IH-335

- (9) Bomb impact locations (X and Y positions for each impact)
- (10) Gun impact locations (X and Y positions for each impact)
- (11) Target control display flag (specifies what target you see in the target channel)
- (12) Background control display flag (specifies what you see in the background channel).
- (13) Weapons control flags:
 - (a) FLOLS on flag (true=on)
 - (b) FLOLS wave off flag (true=wave off)
 - (c) FLOLS auxillary wave off flag (true=wave off)
 - (d) FLOLS cut light flag (true=cut)
 - (e) Viewpoint crash flag (true=viewpoint crashed)
 - (f) Scene freeze flag (true=scene frozen)
 - (g) Bomb impact flag (0=no impact, 1=1 impact, 2=2 impacts)
 - (h) Gun impact flag (0=no impact, 1=1 impact, 2=2 impacts)
 - (i) ASM launch flag (true=launch rockets)
 - (j) SAM launch flag (true=launch SAM)
 - (k) AA active flag (true=flash anti-aircraft pad)
 - (l) AA explode flag (true=explode FLAK)
 - (m) Close hit flag (true=make sky turn transparent yellow)
 - (n) Own hit flag (true=make sky turn solid yellow)
 - (o) IC complete (true=initial condition complete)
 - (p) New command (true=new CIG command is requested)
 - (q) ASM explode (true=make rockets explode)
 - (r) Heads up display select (0, 1, 2 for display 0, 1, 2)

NAVTRAEOUIPCEN IH-335

TABLE 3. SEL TO DEC INTERFACE
 (A) FLOATING POINT PARAMETERS
 32 BIT WORDS

DESCRIPTION		SYMBOL	WORD
VIEWPOINT, FIELD 1	X	ADWVPX	1
POSITION, ATTITUDE	Y	ADWVPY	2
	Z	ADWVPZ	3
	ROLL	ADWVPRL	4
	PITCH	ADWVPPT	5
	YAW	ADWVPYW	6
VIEWPOINT, FIELD 2	DX	ADWVPXD	7
POSITION, ATTITUDE	DY	ADWVPYD	8
	DZ	ADWVPZD	9
	DROLL	ADWVPRLD	10
	DPITCH	ADWVPTD	11
	DYAW	ADWVPYWD	12
MOVING MODEL # 1, FIELD 1	X	ADWM1X	13
POSITION, ATTITUDE	Y	ADWM1Y	14
	Z	ADWM2Y	15
	ROLL	ADWM1RL	16
	PITCH	ADWM1PT	17
	YAW	ADWM1YW	18
MOVING MODEL #1, FIELD 2	DX	ADWM1XD	19
POSITION, ATTITUDE	DY	ADWM1YD	20
	DZ	ADWM1ZD	21
	DROLL	ADWM1RLD	22
	DPITCH	ADWM1PTD	23
	DYAW	ADWM1YWD	24
MOVING MODEL # 2, FIELD 1	X	ADWM2X	25
POSITION, ATTITUDE	Y	ADWM2Y	26
	Z	ADWM2Z	27
	ROLL	ADWM2RL	28
	PITCH	ADWM2PT	29
	YAW	ADWM2YW	30
MOVING MODEL #2, FIELD 2	DX	ADWM2XD	31
POSITION, ATTITUDE	DY	ADWM2YD	32
	DZ	ADWM2ZD	33
	DROLL	ADWM2RLD	34
	DPITCH	ADWM2PTD	35
	DYAW	ADWM2YWD	36

NAVTRAEEQUIPCEN IH-335

TABLE 3(A) (CONTINUED)

DESCRIPTION		SYMBOL	WORD
MOVING MODEL #3	X	ADWM 3X	37
POSITION, ATTITUDE	Y	ADWM 3Y	38
	Z	ADWM 3Z	39
	ROLL	ADWM 3RL	40
	PITCH	ADWM 3PT	41
	YAW	ADWM 3YW	42
MOVING MODEL #3, FIELD 2	DX	ADWM 3XD	43
POSITION, ATTITUDE	DY	ADWM 3YD	44
	DZ	ADWM 3ZD	45
	DROLL	ADWM 3RLD	46
	DPITCH	ADWM 3PTD	47
	DYAW	ADWM 3YWD	48
MOVING MODEL #4	X	ADWM 4X	49
POSITION, ATTITUDE	Y	ADWM 4Y	50
	Z	ADWM 4Z	51
	ROLL	ADWM 4RL	52
	PITCH	ADWM 4PT	53
	YAW	ADWM 4YW	54
MOVING MODEL #4, FIELD 2	DX	ADWM 4XD	55
POSITION, ATTITUDE	DY	ADWM 4YD	56
	DZ	ADWM 4ZD	57
	DROLL	ADWM 4RLD	58
	DPITCH	ADWM 4PTD	59
	DYAW	ADWM 4YWD	60
MOVING MODEL #5	X	ADWM 5X	61
POSITION, ATTITUDE	Y	ADWM 5Y	62
	Z	ADWM 5Z	63
	ROLL	ADWM 5RL	64
	PITCH	ADWM 5PT	65
	YAW	ADWM 5YW	66
MOVING MODEL #5, FIELD 2	DX	ADWM 5XD	67
POSITION, ATTITUDE	DY	ADWM 5YD	68
	DZ	ADWM 5ZD	69
	DROLL	ADWM 5RLD	70
	DPITCH	ADWM 5PTD	71
	DYAW	ADWM 5YWD	72

NAVTRAEEQUIPCEN IH-335

TABLE 3(A) (CONTINUED)

DESCRIPTION		SYMBOL	WORD
MOVING MODEL #6, FIELD 1	X	ADWM6X	73
POSITION, ATTITUDE	Y	ADWM6Y	74
	Z	ADWM6Z	75
	ROLL	ADWM6RL	76
	PITCH	ADWM6PT	77
	YAW	ADWM6YW	78
	DX	ADWM6XD	79
	DY	ADWM6YD	80
	DZ	ADWM6ZD	81
	DROLL	ADWM6RLD	82
	DPITCH	ADWM6PTD	83
	DYAW	ADWM6YWD	84
FLOLS	GLIDE SLOPE ANGLE	ADWFGS	85
	HOOK-TO-EYE ROLL ANGLE	ADWFRA	86
	BRIGHTNESS	ADWFBR	87
	MEATBALL LOCATION	ADWMT	88

NAVTRAEEQUIPCEN IH-335

TABLE 3(A) (CONTINUED)

DESCRIPTION		SYMBOL	WORD
TARGET IMAGE	FIELD 1:	ROLL	ADWTRL1 89
PROJECTOR		PITCH	ADWTPT1 90
		YAW	ADWTYW1 91
	FIELD 2:	ROLL	ADWTRL2 92
		PITCH	ADWTPT2 93
		YAW	ADWTYW2 94
	ZOOM RATIO		ADWTZM 95
TARGET VIEW	FIELD 1:	SINE PITCH	ADWWSP1 96
WINDOW ATTITUDE		COSINE PITCH	ADWWCP1 97
		SINE YAW	ADWWSY1 98
		COSINE YAW	ADWWCY1 99
	FIELD 2:	SINE PITCH	ADWWSP2 100
		COSINE PITCH	ADWWCP2 101
		SINE YAW	ADWWSY2 102
		COSINE YAW	ADWWCY2 103
BOMB IMPACT LOCATIONS	X1	ADWBIX1	104
	Y1	ADWBIY1	105
	X2	ADWBIX2	106
	Y2	ADWBIY2	107
GUNNERY IMPACT LOCATIONS	X1	ADWGIX1	108
	Y1	ADWGIY1	109
	X2	ADWGIX2	110
	Y2	ADWGIY2	111

TABLE 3. (CONTINUED)
 SEL TO DEC INTERFACE
 (8) CONTROL FLAGS

DESCRIPTION		WORD
TARGET CHANNEL CS DISPLAY CONTROL		IDWTCC 112
BACKGROUND CHANNEL CS DISPLAY CONTROL		IDWBCC 113
LDWF ^L FLOLS ON	LDWWO WAVE OFF	LDWWOA AUX WAVE OFF
LDWVC VP CRASH	LDWSF SCENE FREEZE	IDWBIF BOMB IMPACT
LDWR ^L C ASM LAUNCH	LDWSLC SAM LAUNCH	LDWAAA AA ACTIVE
LDWCH CLOSE HIT	LDWOH OWN HIT	LDWICC IC COMPLETE
ASM EXPLORE LDWR ^L E		KDHUD
FIXED MODEL HIT CONTROL		IDWFHT 119
MOVING MODEL HIT CONTROL		IDWMHT 120
REATIME COMMAND BUFFER		IDWNCD 121
		122
		123
		124
		125
		126
		127

- (s) MIL setting for gun sight (1=1 mil, etc.)
- (14) Fixed model hit control (set to 0-15 to make target 0-15 disappear)
- (15) Moving model hit control (set to 1-6 to make moving model 1-6 blow up [only 3 and 4 used])
- (16) Realtime command buffer (set to valid CIG commands)

b. The second task is the transfer of all CIG parameters required for weapon visual software to the appropriate weapon visual parameters. Note that all position parameters are in CIG Earth coordinates (Figure 7). These parameters include DEC status flags, realtime commands, and database target parameters and are shown in Tables 4 and 5. These parameters are specified below:

DEC parameters updated at 30 Hertz:

- (1) Input error code (I/O error: CIG numerical code)
- (2) Buffer error offset (byte offset to first I/O error)
- (3) Buffer error replacement (first value used by CIG to replace, by default, the first erroneous data item found)
- (4) Collision detection flag (true=collision)
- (5) DEC status flag (true=DEC is operating)
- (6) DEC #2 status flag (true=DEC #2 is operating)
- (7) CMD inactive (true=next time big buffer is expected)
- (8) Realtime command buffer (legal CIG command)

DEC parameters updated when new database loaded (one shot):

(1) Active environment set verification (set to integer values 1-32) indicates which database is loaded.

- (a) DAYFORRESTAL = 1
- (b) Standard Meridian = 2
- (c) FLOLS Meridian = 4
- (d) Gunnery Range = 8
- (e) Twin Towns = 16

NAVTRAEEQUIPCEN IH-335

(f) River Valley = 32

(2) Active coordinate sets verification (2-64) indicates which coordinate sets are being used at the time.

- (a) Moving Model 1 = 2
- (b) Moving Model 2 = 4
- (c) Moving Model 3 = 8
- (d) Moving Model 4 = 16
- (e) Moving Model 5 = 32
- (f) Moving Model 6 = 64

(3) Fixed model target control (0-7) indicates the presence of that target in the database loaded.

- (a) There can be up to 8 fixed targets in each database.
- (b) These targets can be deleted by setting the fixed model hit indicator to simulate target destruction.
- (4) Fixed model target identifiers (32 bytes) indicates the name of the target (4 letters).
- (a) Example: tank, towr, etc.
- (5) Fixed model target positions (X, Y, Z positions).
 - (a) There can be a total of eight targets
- (6) Fixed model target kill radius (kill radius in feet)

c. The third task is a test driver. In this mode, all weapon visual software inputs to CIG are bypassed except for those generated by the test driver. The test mode is used for Visual/CIG Interface debugging. The interface buffer size is 512 bytes which is equivalent to 128 SEL words (4 byte words) and 256 DEC words (2 byte words). The test driver has the following two test options:

- (1) The first test option fills the SEL to DEC interface buffer with integers incremented from 1 to 256. This puts an easily identifiable number in each of the 256 DEC (16 bit words) in the buffer which is a handy checkout tool.
- (2) The second test option fills the SEL to DEC interface buffer with a walking bit pattern. Bits 0 through 32 are sequentially set for the first 32 SEL words (32 bit words). This is used for checking bits.

NAVTRAEEQUIPCEN IH-335

TABLE 4. DEC TO SEL INTERFACE
(A) FRAME RATE BUFFER

INPUT ERROR CODE		IDWERR 1		
BUFFER ERROR OFFSET		IDWRPT 2		
BUFFER ERROR REPLACEMENT				IDWRR 3
LDWCOL COLL DETECT	LDWST DEC STATUS (1)	LDWNDB NEW DB	SPARE	LDW SPARE 4 DEC STATUS (2)
CMD INACTIVE	SPARE		5	
REALTIME IDWRSP	COMMAND	RESPONSE	BUFFER	6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18

NAVTRAEEQUIPCEN IH-335

TABLE 5. DEC TO SEL INTERFACE
 (B) DATA BASE INITIALIZATION BUFFER

IDWAES	ACTIVE ENVIRONMENT SET VERIFICATION			19
IDWACS	ACTIVE COORDINATE SETS VERIFICATION			20
IDWFCTC	FIXED MODEL TARGET CONTROL			21
IDWFNTN	FIXED MODEL TARGET IDENTIFIERS			22
ASCE NAME 2				23
3				24
4				25
5				26
6				27
7				28
8				29
FIXED MODEL TARGET 1	POSITION	X	ADWTIX	30
		Y	ADWTIY	31
		Z	ADWTIZ	32
	RADIUS OF VULNERABILITY		ADWTIS	33
FIXED MODEL TARGET 2	X	ADWT2X	34	
	Y	ADWT2Y	35	
	Z	ADWT2Z	36	
	RV	ADWT2S	37	
FIXED MODEL TARGET 3	X	ADWT3X	38	
	Y	ADWT3Y	39	
	Z	ADWT3Z	40	
	RV	ADWT3S	41	
FIXED MODEL TARGET 4	X	ADWT4X	42	
	Y	ADWT4Y	43	
	Z	ADWT4Z	44	
	RV	ADWT4S	45	

NAVTRA EQUIPCEN IH-335

TABLE 5. (CONTINUED)

FIXED MODEL TARGET 5	X	ADWT5X	46
	Y	ADWT5Y	47
	Z	ADWT5Z	48
	RV	ADWT5S	49
FIXED MODEL TARGET 6	X	ADWT6X	50
	Y	ADWT6Y	51
	Z	ADWT6Z	52
	RV	ADWT6S	53
FIXED MODEL TARGET 7	X	ADWT7X	54
	Y	ADWT7Y	55
	Z	ADWT7Z	56
	RV	ADWT7S	57
FIXED MODEL TARGET 8	X	ADWT8X	58
	Y	ADWT8Y	59
	Z	ADWT8Z	60
	RV	ADWT8S	61

NAVTRAEEQUIPCEN IH-335

APPENDIX A
AIRCRAFT CARRIER DRIVER

This subroutine computes the drive parameters for the carrier. Position, heading, velocity, heave commands come from flight module ZM3SNOOA. This subroutine is called by the visual module ZM6SMOVE when the carrier database is loaded at the Experimenter Operator Station (EOS). Figure A-1 illustrates a typical carrier landing approach for the VTRS system. The carrier is a CIG model of the USS Forrestal.

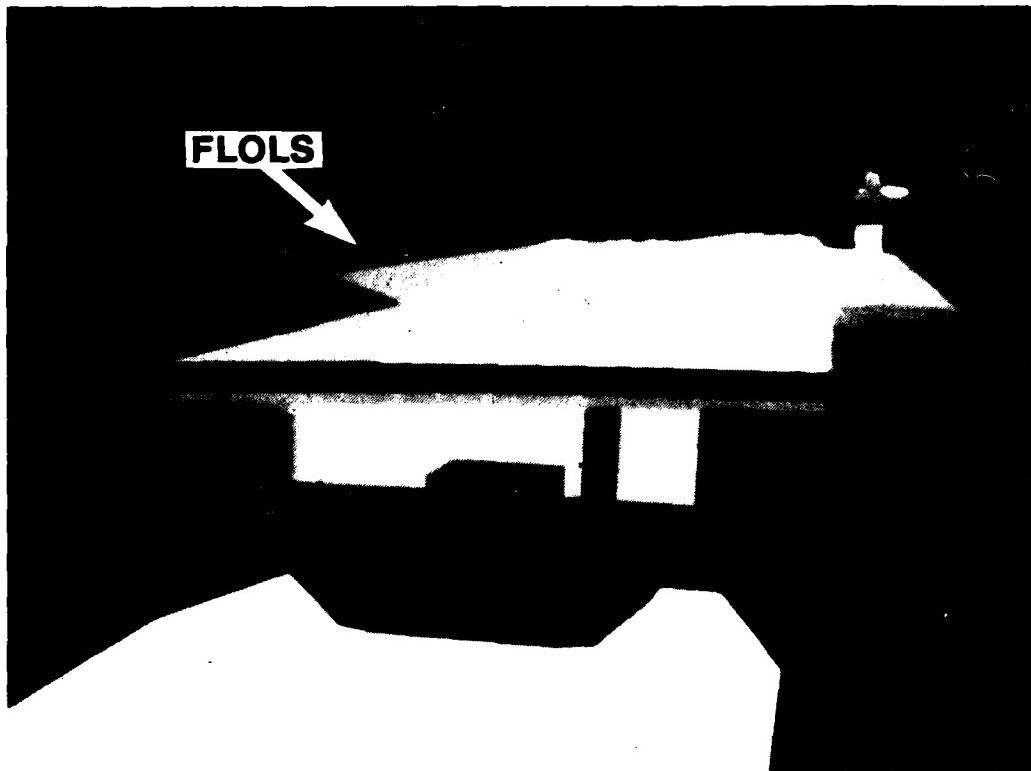


Figure A-1. Carrier Landing Approach on VTRS System.

CARRIER DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The carrier position and attitude are defined relative to the Carrier Reference Point (C.R.P.) in the flight software figure (A-2). Since the visual software needs moving model positions relative to the center of Gravity (C.G.), the following conversion is needed.

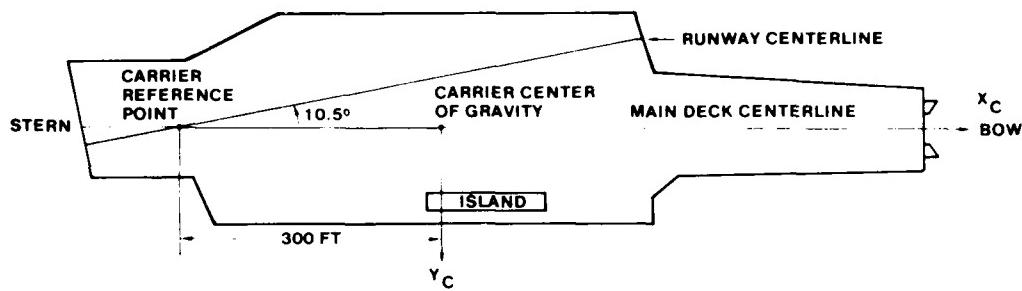


Figure A-2. Carrier Reference Point and Center of Gravity Definition in Carrier Axis System.

Define the X,Y,Z offset between the C.R.P. and the C.G. in the Carrier Axis System:

$$\text{AVXCMGF2}=300.$$

$$\text{AVYCMGF2}=0.$$

$$\text{AVZCMGF2}=0.$$

WHERE: AVXCMGF2, AVYCMGF2, AVZCMGF2 are the X,Y,Z offset between C.R.P. and C.G. in Carrier Axis System.

300. = The X Offset in feet

0. = The Y Offset in feet

0. = The Z Offset in feet

Convert the offset into the CIG Earth Axis System:

$$\text{AVPXOFST}=\text{AVXCMGF2}*\text{AVBTES11}+\text{AVYCMGF2}*\text{AVBTES12}+\text{AVZCMGF2}*\text{AVBTES13}$$

$$\text{AVPYOFST}=\text{AVXCMGF2}*\text{AVBTES21}+\text{AVYCMGF2}*\text{AVBTES22}+\text{AVZCMGF2}*\text{AVBTES23}$$

$$\text{AVPZOFST}=\text{AVXCMGF2}*\text{AVBTES31}+\text{AVYCMGF2}*\text{AVBTES32}+\text{AVZCMGF2}*\text{AVBTES33}$$

WHERE: AVPXOFST, AVPYOFST, AVPZOFST are X,Y,Z offset components in CIG Earth Axis System.

AVXCMGF2, AVYCMGF2, AVZCMGF2 are X,Y,Z offset components in Carrier Axis System.

NAVTRAEEQUIPCEN IH-335

AVBTES11-AVBTES33 represents the target to earth transfer matrix.

Add the X,Y,Z offset components to the carrier position (relative to C.R.P.) to get the C.G. position of the carrier in the CIG Earth Axis System:

$$\begin{aligned} AVWMX(2) &= ANC GEX + AVP X OF ST \\ AVWMY(2) &= ANC GEY + AVP Y OF ST \\ AVWMZ(2) &= ANC GEZ + AVP Z OF ST \end{aligned}$$

WHERE: AVWMX(2), AVWMY(2), AVWMZ(2) are X,Y,Z position of Carrier C.G. in CIG Earth Axis System. (Note that subscript 2 indicates carrier as moving model 2.)
ANC GEX, ANC GEY, ANC GEZ are X,Y,Z position of Carrier C.R.P. in CIG Earth Axis System.
AVP X OF ST, AVP Y OF ST, AVP Z OF ST are X,Y,Z offsets between C.R.P. and C.G. in CIG Earth Axis System.

B. Since there is only a position offset between flight and visual reference points to the carrier, the flight and visual angles correlate one to one and can be passed directly from flight to visual parameters.

Pass flight attitude angles to visual parameters:

$$\begin{aligned} AVWMY(2) &= ANCVAT && \dots \text{Pass Yaw Angle} \\ AVWMPT(2) &= ANTHEC && \dots \text{Pass Pitch Angle} \\ AVWMRL(2) &= ANPHIC && \dots \text{Pass Roll Angle} \end{aligned}$$

WHERE: AVWMRL(2), AVWMPT(2), AVWMY(2) are visual yaw, pitch, roll carrier angles, CIG Earth Axis.
ANPHIC, ANTHEC, ANCVAT are flight yaw, pitch, roll carrier angles, CIG Earth Axis.

C. Next, extrapolations are computed for carrier position attitude. These are required for the CIG System which, runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of carrier position 16.7 milliseconds from now:

$$\begin{aligned} AVWMXD(2) &= 1.5 * AVWMX(2) - .5 * AVWMX1 \\ AVWMYD(2) &= 1.5 * AVWMY(2) - .5 * AVWMY1 \\ AVWMZD(2) &= 1.5 * AVWMZ(2) - .5 * AVWMZ1 \end{aligned}$$

WHERE: AVWMXD(2), AVWMYD(2), AVWMZD(2) are extrapolated X, Y, Z carrier position, CIG Earth Axis System.

AVWMX(2), AVWMY(2), AVWMZ(2) are current X, Y, Z carrier position, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z carrier position,
CIG Earth Axis System.

Save the previous frame position:

```
AVWMX1=AVWMX(2)  
AVWMY1=AVWMY(2)  
AVWMZ1=AVWMZ(2)
```

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frames carrier X, Y, Z position, CIG
Earth Axis System.
AVWMX(2), AVWMY(2), AVWMZ(2) are current carrier X, Y, Z position,
CIG Earth Axis System.

Compute the extrapolated value of carrier attitude 16.7 milliseconds
from now:

```
AVWMYWD(2)=1.5*AVWMYW(2)-.5*AVWMYW1  
AVWMPTD(2)=1.5*AVWMPT(2)-.5*AVWMPT1  
AVWMRLD(2)=1.5*AVWMRL(2)-.5*AVWMRL1
```

WHERE: AVWMYWD(2), AVWMPTD(2), AVWMRLD(2) are extrapolated yaw, pitch, roll
angles, CIG Earth Axis.
AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles,
CIG Earth Axis.
AVWMYW1, AVWMPT1, AVWMRL1 are last frames yaw, pitch, roll angles,
CIG Earth Axis.

Save last frame attitude values:

```
AVWMYW1=AVWMYW(2)  
AVWMPT1=AVWMPT(2)  
AVWMRL1=AVWMRL(2)
```

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frames yaw, pitch, roll angles,
CIG Earth Axis System.
AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles,
CIG Earth Axis System.

D. The input module, output module, and parameters used by this
subroutine are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request
(2) ZM3SN00A - Inputs carrier (C.R.P.) position,
attitude

Output Module: (2) ZM6SWCGI - CVA X, Y, Z Pos CIG Earth, and
extrapolations
- CVA PT, YW, RL Ang, CIG Earth, and
extrapolations

APPENDIX B
FLAK DRIVER SUBROUTINE

This module computes the drive parameters for anti-aircraft flak. This subroutine is called by the visual module ZM6SMOVE when either the Twin Town or River Valley database is loaded and the flak launch flag LJBFLAK is set true at the experimenter operator station (EOS). Figure B-1 illustrates a typical flak burst for the River Valley database.

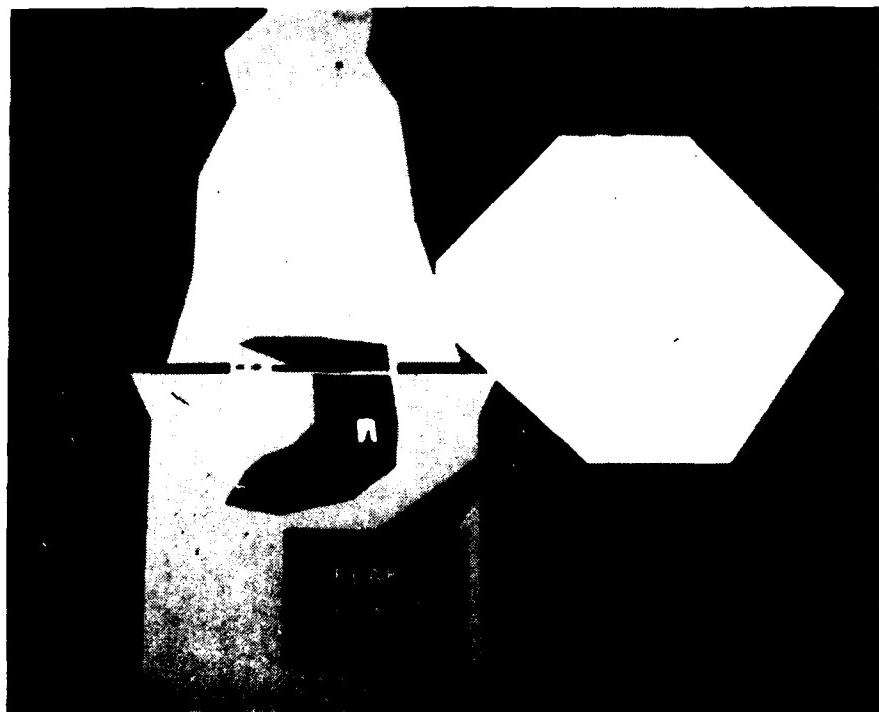


Figure B-1. Anti-Aircraft FLAK Burst.

FLAK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The first step in the flak driver algorithm is to initialize the flak launch parameters. This includes placement of the flak on the launch pad (Figure B-2) and resetting flak timers and hit flags. As shown in Figure B-2, the flak is placed at the center of the launch pad.

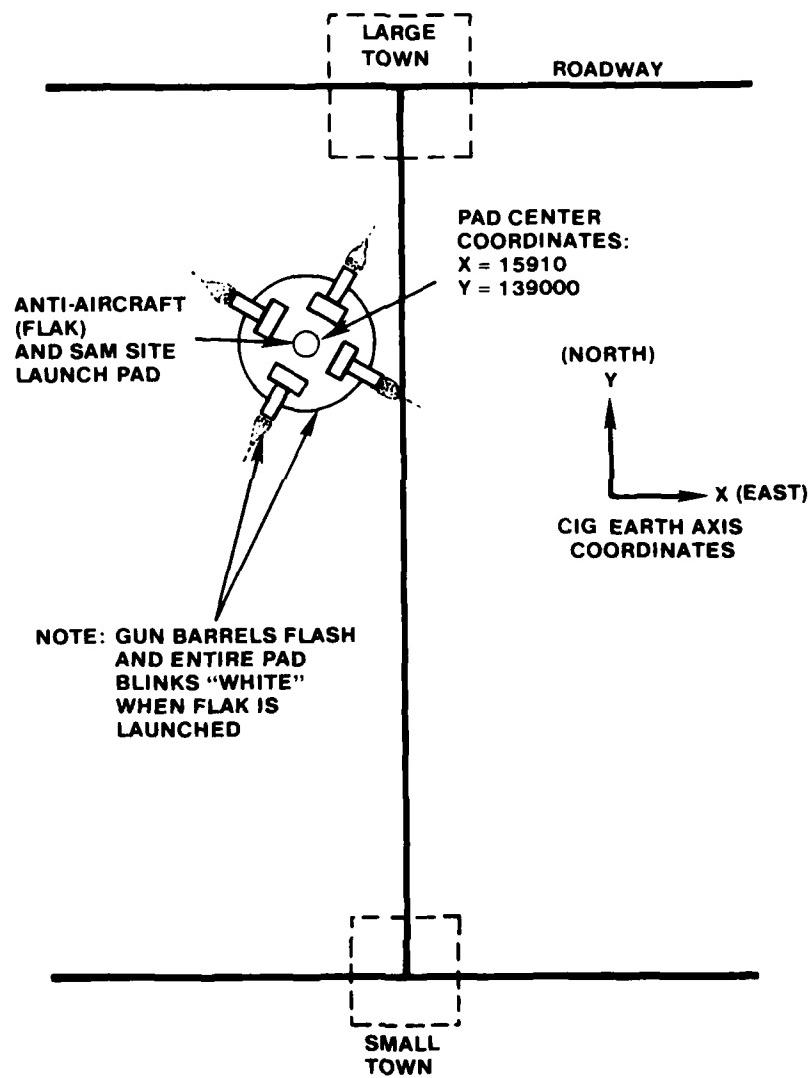


Figure B-2. Location of FLAK and SAM Launch Pad in CIG Earth Axis.

NAVTRAEEQUIPCEN IH-335

Initialize flak starting position:

AJFLAKX=AJFLAKXS ...PAD X position coordinate = 15,910 feet
AJFLAKY=AJFLAKYS ...PAD Y position coordinate = 139,000 feet
AJFLAKZ=AJFLAKZS ...PAD Z position coordinate = 0 feet

Initialize flak attitude angles:

AJFLAKYW=0. ...Yaw = 0 degrees
AJFLAKPT=90. ...Pitch = 90 degrees
AJFLAKRL=0. ...Roll = 0 degrees

WHERE: AJFLAKX, AJFLAKY, AJFLAKZ are flak X, Y, Z current position, CIG Earth Axis.
AJFLAKXS, AJFLAKYS, AJFLAKZS are X, Y, Z launch position of flak, CIG Earth Axis System.
AJFLAKYW, AJFLAKPT, AJFLAKRL are current flak yaw, pitch, roll angles.

Flak launch flags and timers are initialized:

LJBFLAK=.FALSE.
LJBFLAKFLT=.TRUE.
LJWCH=.FALSE.
LJWOH=.FALSE.
LJWAEE=.FALSE.
IJFLAKHT=0
IJFLAKTM=0
IJFTIME=60
IJFBURST=0

WHERE: LJFLAKFLT = True When Flak is in Flight (Active)
LJBFLAK = True to Launch Flak (Set False After First Pass)
LJWCH = Flak Passed Within 100 Ft When = True (Close Hit)
LJWOH = Flak Passed Within 50 Ft When = True (Own Hit)
LJWAEE = Flak Explode Flag For New Hit
IJFLAKHT = Flak Flight Timer, 30 Per Second
IJFLAKHTM = Close/Own Hit Timer, 30 Per Second
IJFTIME = Flak Burst Time = 60 For 2 Second Burst (60 program loops = 2 seconds)
IJFBURST = Flak Burst Counter = Initialized to 0 to Start

B. The second step is to flash the flak launch pad for 5 seconds. This is done to simulate the acquisition phase of a flak launch. The flashing pad indicates to the pilot that the flak site has locked on to the T-2C and is preparing to launch a flak against him.

NAVTRAEEQUIPCEN IH-335

Let pad flash for 5 seconds, then go to flak trajectory phase:

LJWSLC=.TRUE.	... Flash Flak Launch Pad
IJFLAKTM=IJFLAKTM+1	... Increment Flak Flight Counter
IF(IJFLAKTM.LT.150) GO TO 20	... Bypass Flak Trajectory for First 150 Loops (5 Seconds)
LJWAAA=.TRUE.	... Make AAA Gun Barrels Flash
LJWSLC=.TRUE.	... Make AAA Pad Flash
20 CONTINUE	... Pass Flak Launch Position
LJWAAA=.FALSE.	... Turn Off AAA Gun Flash
LJWSLC=.FALSE.	... Turn Off Launch Pad Flash

WHERE: IJFLAKTM is flak subroutine timer.

LJWAAA = True to Flash AAA Gun Barrels

LJWSLC = True to Flash AAA Launch Pad (So Pilot Can See It)

150 = 5 Seconds of Pad Flash = 150 Program Loops

C. The third step is to compute the flak trajectory and explosions. Six flak bursts are randomly put in front of the pilot's aircraft at two second intervals. The distance in front of the aircraft and the radius of the dispersion pattern for the flak can be controlled at the EOS. Figure B-3 shows a typical flak pattern. Flak bursts will continue for 12 seconds (each burst lasts 2 seconds) unless pilot is hit (burst within 10 feet) or pulls High G evasive maneuver (200 deg/sec in yaw, pitch, or roll). The screen will flash a transparent yellow for a near hit (burst within 50 feet) or a solid yellow for a hit.

Set up flak timer for 2 second flak burst:

IJFTIME=IJFTIME + 1	.. Increment Burst Timer
IF(IJFTIME.LT.60) GO TO 200	.. Let Each Flak Burst Stay Put For 2 Seconds (Lets Aircraft Fly By Flak Bursts Suspended In Air)

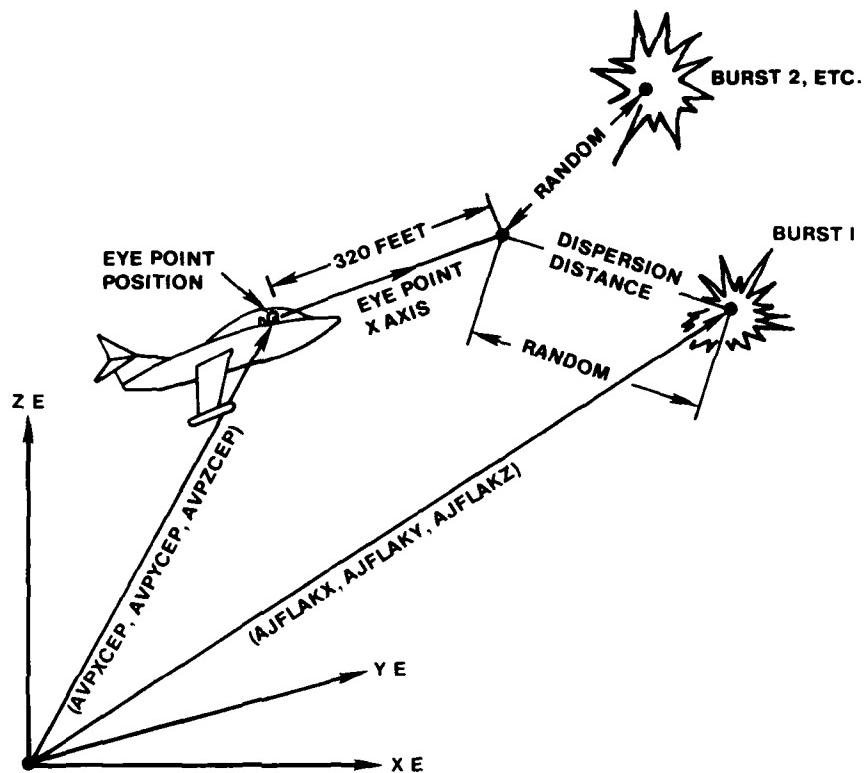
WHERE: IJFTIME=60 = 2 Seconds of Flak Burst = Burst Counter

60 = 2 Seconds (2 * 30 Program Loops)

200 Is Entry Point For Next Flak Position

Set up flak reference point in body axis system:

AJFLAKXS=320.	.. Put Reference Point for Flak Position 320 Feet in Front of Pilot's EyePoint (320 Feet Out +X Axis)
AJFLAKYS=0.	
AJFLAKZS=0.	



SIX RANDOM FLAK POSITIONS ARE COMPUTED AT 2 SECOND INTERVALS. EACH FLAK POSITION REMAINS FIXED IN SPACE WHILE AIRCRAFT FLIES AROUND/THROUGH EXPLOSION. THE DISPERSION DISTANCE CAN BE CONTROLLED AT THE EOS TO INCREASE/DECREASE PILOT AVOIDANCE TASK IF PILOT HITS FLAK (PASSES WITHIN 10 FEET), HE GETS SOLID YELLOW FIREBALL ON VISUAL SCREEN.

Figure B-3. FLAK Burst Position in CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

WHERE: AJFLAKXS, AJFLAKYS, AJFLAKZS is flak reference point, CIG Aircraft Axis System.

320. = +320 Feet in the Aircraft X Axis

0 = 0 Feet in Aircraft Y Axis

0 = 0 Feet in Aircraft Z Axis

Get flak reference point in earth axis system.

AJFLAKXE=AJFLAKXS*AVBDER11+AJFLAKYS*AVBDER12+AJFLAKZS*AVBDER13

AJFLAKYE=AJFLAKXS*AVBDER21+AJFLAKYS*AVBDER22+AJFLAKZS*AVBDER23

AJFLAKZE=AJFLAKXS*AVBDER31+AJFLAKYS*AVBDER32+AJFLAKZS*AVBDER33

Compute random X, Y, Z position components for flak and add them to flak reference point (320 feet in front of eyepoint):

AJFLAKX=AVPXCEP + AJFLAKXE + ZARDMA*AJVFMITS

AJFLAKY=AVPYCEP + AJFLAKYE + ZARDMA*AJVFMITS

AJFLAKZ=AVPZCEP + AJFLAKZE + ZARDMA*AJVFMITS

WHERE: AJFLAKX, AJFLAKY, AJFLAKZ are random X, Y, Z flak positions, CIG Earth Axis System.

AVPXCEP, AVPYCEP, AVPZCEP are X, Y, Z eyepoint positions, CIG Earth Axis System.

AJFLAKXE, AJFLAKYE, AJFLAKZE are X, Y, Z flak reference point positions, CIG Earth Axis.

ZARDMA is a random decimal number which varies between 0 and 1 (produced by random number generator).

AJVFMITS is scale factor for random flak burst positions (this controls magnitude of dispersion pattern).

The timer for the flak burst (length of each explosion) is reset and flak counter (number of flak bursts) is incremented:

IJFTIME=0

IJFBURST=IJFBURST + 1

WHERE: IJFTIME is length of flak bursts (6 for 2 seconds).

IJFBURST is number of flak bursts.

Next, scoring is determined for flak explosions. If the pilot evades 200 deg/sec in yaw, pitch, or roll; or the flak misses by 50 feet, then no hit is scored. If the flak miss is less than 50 feet a close hit occurs. If the flak miss is less than 10 feet, a direct hit is scored:

200 CONTINUE

IF(IJFLAKTM.GT.510) GO TO 225

.. Entry Point for Flak Scoring

.. Flak Driver Out of Time

Reset parameters and Exit

NAVTRAEEQUIPCEN IH-335

Let flak miss if pilot evades 200 deg/sec in any attitude:

IF(AVFDPA.GT.200) GO TO 250 .. Roll Exceeds 200 Deg/Sec
IF(AVFDQA.GT.200) GO TO 250 .. Yaw Exceeds 200 Deg/Sec
IF(AVFDRA.GT.200) GO TO 250 .. Pitch Exceeds 200 Deg/Sec

WHERE: AVFDPA, AVFDQA, AVFDRA are aircraft roll, yaw, pitch rate in deg/sec CIG Body Axis.

IJFLAKTM is FLAK subroutine counter (increments once per program loop)

510 = 5 seconds pad flash + 6 flak bursts (2 seconds each) =
 $(5 + 6 \times 2) \times 30 = 510$ loops

250 = Program entry point beyond flak hit/near hit code

Compute flak miss distance:

AJXMIS=AJFLAKX-AVPXCEP

AJYMIS=AJFLAKY-AVPYCEP

AJZMIS=AJFLAKZ-AVPZCEP

AJVFMIS=SQRT(AJXMIS*AJXMIS+ AJYMIS*AJYMIS+ AJZMIS*AMZMIS)

WHERE: AJXMIS, AJYMIS, AJZMIS are X, Y, Z flak miss distance components in CIG Earth Axis System.

AJFLAKX, AJFLAKY, AJFLAKZ are X, Y, Z flak positions in CIG Earth Axis System.

AVPXCEP, AVPYCEP, AVPZCEP are X, Y, Z eyepoint positions in CIG Earth Axis System.

AJVFMIS is vector magnitude of flak miss distance.

Check for close hit and direct hit. If a close hit occurs, set close hit flag. If a direct hit occurs set direct hit flag and exit program after 5 seconds of solid yellow on screen:

IF(AJVFMIS.LT.50.) LJWCH=.TRUE. .. If Close Hit, Set Flag True (transparent yellow on screen)
IF(AJVFMIS.GT.10.) GO TO 250 .. No Direct Hit, So Keep Shooting Flak
LJWOH=.TRUE. Set "HIT" TRUE .. Direct Hit, So Kill Pilot (Solid Yellow on Screen)

WHERE: AJVFMIS is magnitude of flak miss distance

LJWCH is close hit flag

LJWOH is hit flag

50 = 50 feet miss distance for close hit

10 = 10 feet miss distance for direct hit

NAVTRAEEQUIPCEN IH-335

Now the flak trajectory is complete, so the flak is now put out in the south forty and its attitude reset:

```
225 CONTINUE
    AJFLAKX=99999999.          .. Flak Termination Entry Point
                                .. Put X, Y, Z Flak Position in South Forty
                                (removes flak from scene)
    AJFLAKY=99999999.
    AJFLAKZ=99999999.

    AJFLAKPT=90.0             .. Reset Flak Yaw, Pitch, Roll Angles
    AJFLAKYW=0.0
    AJFLAKRL=0.0
```

Turn off AA explode, close hit, own hit, flak flight flags. Let flak hit flags persist for 5 seconds:

```
IJFLAKHT=IJFLAKHT+1
IF(IJFLAKHT.LT.150) GO TO 250      .. Increment Hit Timer
LJWAAE=.FALSE.                      .. Let Hits Persist for 5 Sec
LJWCH=.FALSE.                       .. Reset AA Explode
LJWOH=.FALSE.                       .. Reset Close Hit Flag
LJFLKFLT=.FALSE.                    .. Reset Own Hit Flag
                                      .. Reset Flak Flight Flag
```

WHERE: IJFLAKHT is Flak Hit Timer = 150 for 5 Seconds
LJWAAE is Flak Explode Flag = True for Explosion
LJWCH is Flak Close Hit Flag = True for Close Hit
LJWOH is Flak Own (Direct) Hit = True for Direct Hit
LJFLKFLT is Flak Flight Timer = True for Active Flak
250 = program address that bypasses flak reset code

D. Next, extrapolations are computed for flak position attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

First, the flak explode timer is checked and turned off after position.

Compute the extrapolated value of flak position 16.7 milliseconds from now:

```
250 CONTINUE
    Pass flak position to visual:
```

```
AVWMX(5)=AJFLAKX
AVWMY(5)=AJFLAKY
AVWMZ(5)=AJFLAKZ

AVWMXD(5)=1.5*AVWMX(5) - .5*AVWMX1
AVWMD(5)=1.5*AVWMY(5) - .5*AVWY1
AVWMZD(5)=1.5*AVWMZ(5) - .5*AVWMZ1
```

NAVTRAEEQUIPCEN IH-335

WHERE: AVWMXD(5), AVWMYD(5), AVWMZD(5) are extrapolated X, Y, Z flak position, CIG Earth Axis System.
AVWMX(5), AVWMY(5), AVWMZ(5) are current X, Y, Z flak position, CIG Earth Axis System.
AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z flak position, CIG Earth Axis System.

Save the previous frame position.

AVWMX1=AVWMX(5)
AVWMY1=AVWMY(5)
AVWMZ1=AVWMZ(5)

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frame's flak X, Y, Z position, CIG Earth Axis System.
AVWMX(5), AVWMY(5), AVWMZ(5) are current flak X, Y, Z position, CIG Earth Axis System.

Pass flak attitude angles to visual:

AVWMYW(5)=AJFLAKYW
AVWMPT(5)=AJFLAKPT
AVWMRL(5)=AJFLAKRL

Compute the extrapolated value of flak attitude 16.7 milliseconds from now:

AVWMYWD(5)=1.5*AVWMYW(5) - .5*AVWMYW1
AVWMPTD(5)=1.5*AVWMPT(5) - .5*AVWMPT1
AVWMRLD(5)=1.5*AVWMRL(5) - .5*AVWMRL1

WHERE: AVWMYWD(5), AVWMPTD(5), AVWMRLD(5) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW(5), AVWMPT(5), AVWMRL(5) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMYW1=AVWMYW(5)
AVWMPT1=AVWMPT(5)
AVWMRL1=AVWMRL(5)

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
AVWMYW(5), AVWMPT(5), AVWMRL(5) are current yaw, pitch, roll angles, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

E. The input module, output module, and input/output parameters are listed below:

Input Module: (1) ZM6SWCGI - Flak X, Y, Z Start Pos, CIG Earth
(2) ZM6SW033 - X, Y, Z Eyept Pos, CIG Earth
- AC Attitude acceleration angles (yaw,
pitch, roll)
- Flak Launch, T/F

Output Module: (1) ZM6SWCGI - Flak X, Y, Z Pos, CIG Earth,
extrapolations
- Flak Pt, Yw, Rl, CIG Earth,
extrapolations
- Launch, Close Hit, Hit, T/F

APPENDIX C

FLOLS DRIVER

This subroutine computes the position and attitude of the fresnal lens optical landing system (FLOLS) and the FLOLS meatball position. The FLOLS is a landing aid used by the pilot to stay at the correct glideslope (elevation) angle when making landings on either the carrier or FLOLS Meridian databases. This subroutine is automatically called by the module ZM6SMOVE when one of the above data bases is loaded. Figure A-1 shows the CIG model of the FLOLS (black box with 2 horizontal white lines and white dot midway between the 2 lines) located to the left of the aircraft carrier runway. The two white lines represent the FLOLS datum line and the white dot is the FLOLS meatball. The pilot is on correct glideslopes when the meatball is level with the datum line. If the meatball is displaced upwards by a distance equal to twice the diameter of the meatball, the pilot is "2 balls" high (3/4 degree above glideslope). An equal negative displacement would be 2 balls low. The pilot must be within 2 balls displacement for a safe landing.

FLOLS DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION.

A. First, the FLOLS position and attitude is determined. There are two situations that must be addressed: (1) Aircraft Carrier FLOLS; (2) Meridian Airport FLOLS.

1. The FLOLS position and attitude for the Forrestal Aircraft Carrier is illustrated in Figure C-1. These parameters are defined in the carrier axis system and transformed into the CIG Earth Axis System:

AVPXFLCM=-28.0 feet	...Define FLOLS position in
AVPYFLCM=-134.834 feet	carrier axis system
AVPZFLCM=-63.028 feet	

$$\begin{aligned} \text{AVPXFLGT} &= \text{AVPXFLCM} * \text{AVBETS11} + \text{AVPYFLCM} * \text{AVBETS12} + \text{AVPZFLCM} * \text{AVBETS13} \\ \text{AVPYFLGT} &= \text{AVPXFLCM} * \text{AVBETS21} + \text{AVPYFLCM} * \text{AVBETS22} + \text{AVPZFLCM} * \text{AVBETS23} \\ \text{AVPZFLGT} &= \text{AVPXFLCM} * \text{AVBETS31} + \text{AVPYFLCM} * \text{AVBETS32} + \text{AVPZFLCM} * \text{AVBETS33} \end{aligned}$$

WHERE: AVPXFLGT, AVPYFLGT, AVPZFLGT are FLOLS X, Y, Z position offsets from carrier C.G., CIG Earth Axis.
 AVPXFLCM, AVPYFLCM, AVPZFLCM are FLOLS X, Y, Z position offsets from carrier C.G., Carrier Axis.
 AVBETS11-33 are target to earth matrix elements.

Next, the above offsets are added to the carrier position in CIG Earth Axis System to get the FLOLS position:

$$\begin{aligned} \text{AVWMX}(5) &= \text{AVWMX}(2) + \text{AVPXFLGT} \\ \text{AVWMY}(5) &= \text{AVWMY}(2) + \text{AVPYFLGT} \\ \text{AVWMZ}(5) &= \text{AVWMZ}(2) + \text{AVPZFLGT} \end{aligned}$$

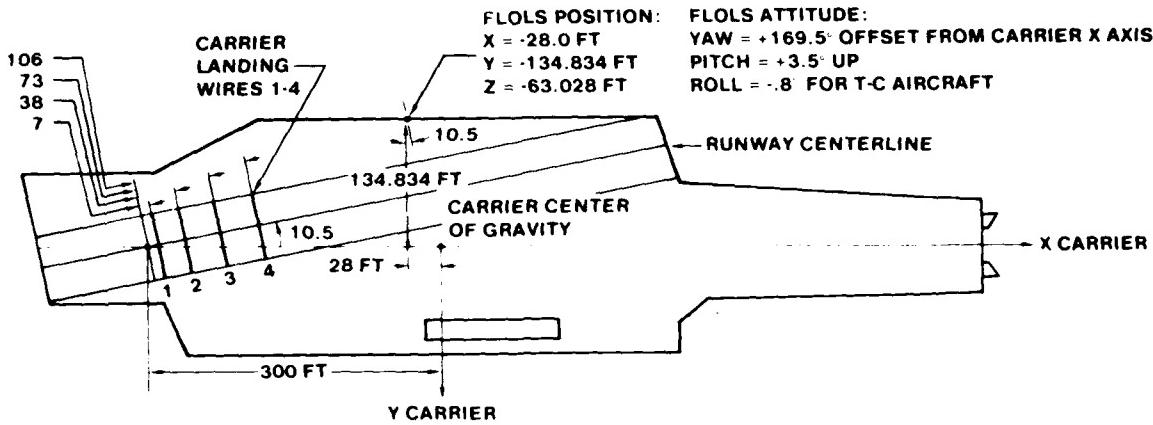


Figure C-1. FLOLS Position on Carrier in Carrier Axis System.

WHERE: AVWMX(5), AVWMY(5), AVWMZ(5) are X, Y, Z FLOLS positions in CIG Earth Axis System. (Note: The CIG system presently ignores these parameters and the FLOLS position on the ship is controlled by the CIG system.)
AVWMX(2), AVWMY(2), AVWMZ(2) are X, Y, Z carrier positions in CIG Earth Axis System.
AVPXFLGT, AVPYFLGT, AVPZFLGT are X, Y, Z FLOLS offset from carrier in CIG Earth Axis.

Compute the FLOLS attitude in the CIG earth axis system:

AVWMRL(5)=AVWMRL(2)	.. Let FLOLS Image Roll With Carrier
AVWMPT(5)=AVWMPT(2)+AVFLSELV	.. Pass FLOLS Elevation
AVWMYW(5)=AVWMYW(2)-10.5	.. Align FLOLS with carrier runway (rotated -10.5 degrees from carrier)

WHERE: AVWMRL(5), AVWMPT(5), AVWMYW(5) are FLOLS roll, pitch, yaws, CIG Earth Axis System. (Note: In the carrier database, these parameters are ignored by the CIG - it controls the FLOLS attitude.)

AVFLSELV is FLOLS elevation angle, CIG Earth Axis.
 -10.5 is FLOLS yaw offset from carrier, CIG Earth Axis.

2. The FLOLS position and attitude for the FLOLS Meridian Airport is illustrated in Figure C-2. Note that the pitch and roll angles for the moving model inputs are different than the pitch and roll angles used for meatball calculations. This makes the FLOLS box appear upright (no roll, etc.) along the runway. Separate code is used to compute the meatball position which uses FLOLS roll and elevation angles to keep aircraft on correct glideslope. These parameters are defined in the CIG Earth Axis System, since the airport is defined in the CIG Earth Axis System.

AVWMX(5)=AVPXFLCW	.. X Position Input (-4170 Ft)
AVWMY(5)=AVPYFLCW	.. Y Position Input (-8295 Ft)
AVWMZ(5)=AVPZFLCW	.. Z Position Input (4.5 Ft)

WHERE: AVWMX(5), AVWMY(5), AVWMZ(5) are X, Y, Z FLOLS positions in CIG Earth Axis System.
 AVPXFLCW, AVPYFLCW, AVPZFLCW are X, Y, Z FLOLS positions variable inputs, CIG Earth Axis.

Get FLOLS yaw, pitch, roll angles for Meridian Airport:

AVWMRL(5)=AVFLSROL=0.	.. Roll Angle Input (Deg)
AVWMPT(5)=AVFLSELV=0.	.. Pitch Angle Input (Deg)
AVWMYW(5)=AVFLSHDG=8.864	.. Yaw Angle Input (Deg)

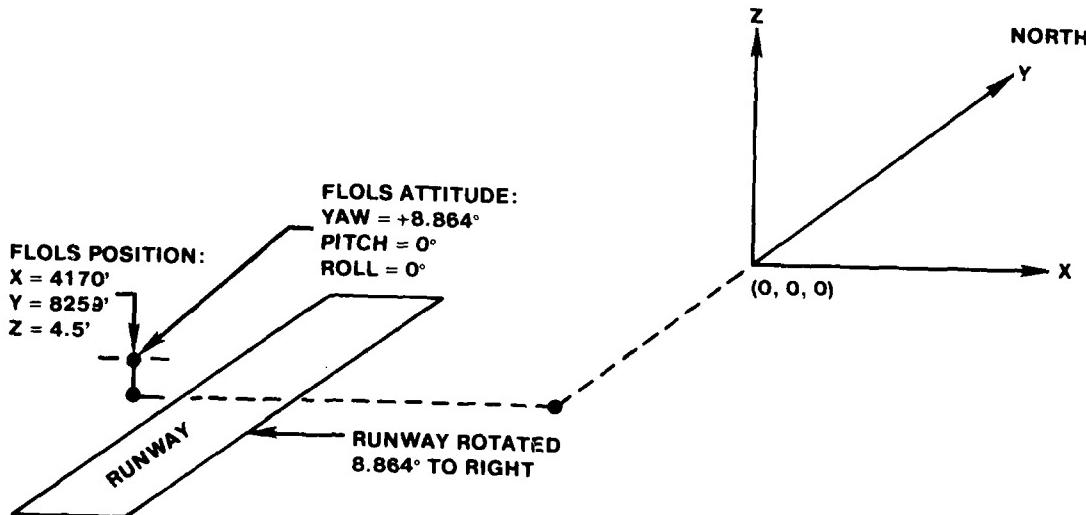


Figure C-2. FLOLS Position and Attitude for Meridian Airport Data Base.

NAVTRAEEQUIPCEN IH-335

WHERE: AVWMY(5), AVWMPT(5), AVWMRL(5) are yaw, pitch, roll angles, CIG Earth Axis System.
AVFLSROL, AVFLSELV, AVFLSHDG are yaw, pitch, roll angle inputs for FLOLS in CIG Earth Axis.

B. Next, extrapolations are computed for FLOLS position attitude. These are required for the CIG System which, runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of FLOLS position 16.7 milliseconds from now:

$$\begin{aligned} \text{AVWMXD}(5) &= 1.5 * \text{AVWMX}(5) - .5 * \text{AVWMX1} \\ \text{AVWMD}(5) &= 1.5 * \text{AVWMY}(5) - .5 * \text{AVWMY1} \\ \text{AVWMZD}(5) &= 1.5 * \text{AVWMZ}(5) - .5 * \text{AVWMZ1} \end{aligned}$$

WHERE: AVWMXD(5), AVWMD(5), AVWMZD(5) are extrapolated X, Y, Z FLOLS position, CIG Earth Axis System.
AVWMX(5), AVWMY(5), AVWMZ(5) are current X, Y, Z FLOLS position, CIG Earth Axis System.
AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z FLOLS position, CIG Earth Axis System.

Save the previous frame position:

$$\begin{aligned} \text{AVWMX1} &= \text{AVWMX}(5) \\ \text{AVWMY1} &= \text{AVWMY}(5) \\ \text{AVWMZ1} &= \text{AVWMZ}(5) \end{aligned}$$

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frame's FLOLS X, Y, Z position, CIG Earth Axis System.
AVWMX(5), AVWMY(5), AVWMZ(5) are current FLOLS X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of FLOLS attitude 16.7 milliseconds from now:

$$\begin{aligned} \text{AVWMD}(5) &= 1.5 * \text{AVWMY}(5) - .5 * \text{AVWMY1} \\ \text{AVWMPTD}(5) &= 1.5 * \text{AVWMPT}(5) - .5 * \text{AVWMPT1} \\ \text{AVWMRLD}(5) &= 1.5 * \text{AVWMRL}(5) - .5 * \text{AVWMRL1} \end{aligned}$$

WHERE: AVWMD(5), AVWMPTD(5), AVWMRLD(5) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.

NAVTRAEEQUIPCEN IH-335

AVWMYW(5), AVWMPT(5), AVWMRL(5) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMYW1=AVWMYW(5)
AVWMPT1=AVWMPT(5)
AVWMRL1=AVWMRL(5)

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw pitch, roll angles, CIG Earth Axis System.
AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis System.

C. Next, the FLOLS landing parameters are computed. These include:
(1) FLOLS control flags; (2) FLOLS meatball position computation.

1. First, the FLOLS control flags are determined. The control flags drive the FLOLS cut lights, wave off lights, and auxillary cut lights. These lights are used by the Navy Landing Signal Control Officer (LSO) to issue landing commands to the pilot. For example, to abort a landing attempt, the wave off lights would be turned on by the LSO.

Check the FLOLS control flags set at the experimenter operator station (EOS) and pass their values to the visual software:

LVCUTLGT = LWVCTLGT
LVAUXLGT = .NOT. LWVAXLGT
LVWAVLGT = LWVWVLGT

WHERE: LVCUTLGT, LWVCTLGT are visual and EOS FLOLS cut light control flags (respectively).
LVAUXLGT, LWVAXLGT are visual and EOS FLOLS auxillary wave off control flags (respectively).
LVWAVLGT, LWVWVLGT are visual and EOS FLOLS wave off control flags (respectively).

2. Second, the FLOLS meatball drive is computed. The meatball angle defines the position of the FLOLS meatball in reference to the FLOLS datum line (see Figure C-3). When the meatball is in the center of the datum line, the pilot is exactly on glideslope. If the pilot is 3/4 of a degree high or low, the meatball will be displaced by 2 balls (its maximum displacement) up or down, respectively. The meatball angle is derived below:

NAVTRAEEQUIPCEN IH-335

Compute the eyepoint to FLOLS vector in CIG Earth Axis:

$$AVXEEPFL = AVPXCEP - AVWMX(5)$$

$$AVYEEPFL = AVPYCEP - AVWMY(5)$$

$$AVZEEPFL = AVPZCEP - AVWMZ(5)$$

WHERE: AVXEEPFL, AVYEEPFL, AVZEEPFL are X, Y, Z components of eyepoint to FLOLS vector, CIG Earth Axis System.
 AVPXCEP, AVPYCEP, AVPZCEP are X, Y, Z components of eyepoint vector, CIG Earth Axis.
 AVWMX(5), AVWMY(5), AVWMZ(5) are X, Y, Z components of FLOLS position, CIG Earth Axis System.

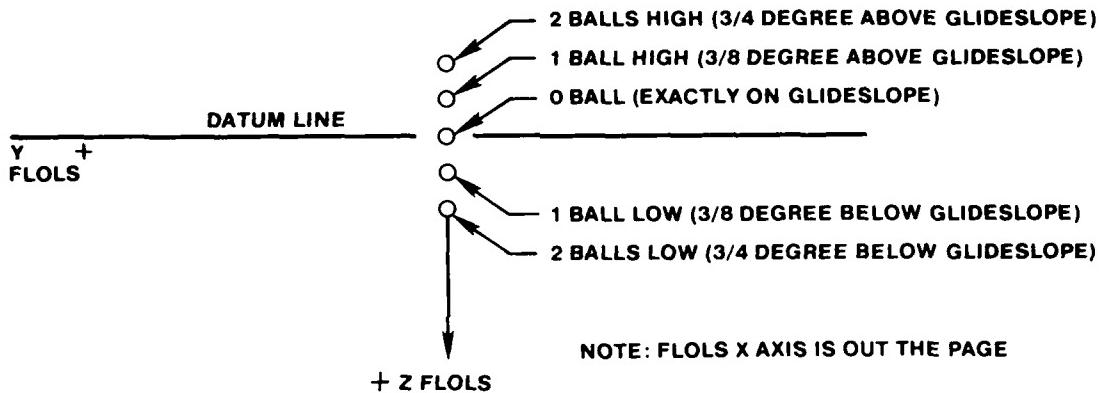


Figure C-3. FLOLS Meatball Position Relative to Aircraft Glideslope.

Next, the sine and cosine of FLOLS yaw, pitch, and roll are computed so that the Earth to FLOLS transfer matrix can be determined:

$$AVFLSHDG=(AVWMYW(5)+180.)$$

- .. Convert CIG FLOLS Yaw to True FLOLS Yaw (FLOLS X axis must swing around 180 degrees to point at aircraft)
- .. Get Sine of FLOLS Yaw
- .. Get Cosine of FLOLS Yaw
- .. Get Sine of FLOLS Roll
- .. Get Cosine of FLOLS Roll
- .. Get Sine of FLOLS Pitch
- .. Get Cosine of FLOLS Pitch

```

AVSIN1=SIN(AVFLSHDG)
AVCOS1=COS(AVFLSHDG)
AVSIN2=SIN(AVFLSROL)
AVCOS2=COS(AVFLSROL)
AVSIN3=SIN(AVFLSELV)
AVCOS3=COS(AVFLSELV)

```

NAVTRAEEQUIPCEN IH-335

WHERE: AVFLSHDG, AVFLSELV, AVFLSROL are true FLOLS yaw, pitch, roll angles in degrees
AVWMYW(5) is FLOLS yaw modified for CIG FLOLS Model, since the backside must face pilot, it is 180 degrees out of phase with true FLOLS yaw.
AVWMRL(5) is FLOLS roll angle, CIG Earth Axis.
AVWMPT(5) is FLOLS pitch angle, CIG Earth Axis.

The earth to FLOLS axis transfer matrix is now defined: In the standard right-handed system (it will have to be converted to the CIG Earth Axis Coordinate System in the next step).

```
AVSEFL11 = AVCOS3 * AVCOS1
AVSEFL12 = AVCOS3 * AVSIN1
AVSEFL13 = -AVSIN3
AVSEFL21 = AVSIN2 * AVSIN3 * AVCOS1 - AVCOS2 * AVSIN1
AVSEFL22 = AVSIN2 * AVSIN3 * AVSIN1 + AVCOS2 * AVCOS1
AVSEFL23 = AVSIN2 * AVCOS3
AVSEFL31 = AVCOS2 * AVSIN3 * AVCOS1 + AVSIN2 * AVSIN1
AVSEFL32 = AVCOS2 * AVSIN3 * AVSIN1 - AVSIN2 * AVCOS1
AVSEFL33 = AVCOS2 * AVCOS3
```

WHERE: AVSIN1, AVCOS1 are sine and cosine of FLOLS yaw.
AVSIN2, AVCOS2 are sine and cosine of FLOLS roll.
AVSIN3, AVCOS3 are sine and cosine of FLOLS pitch.
AVSEFL11-33 are earth to FLOLS matrix elements in a standard right-handed system.

The earth axis to FLOLS is now converted into the CIG Earth Axis System (X and Y axis interchanged, Z axis inverted):

```
AVERFL11= AVSEFL21
AVERFL22= AVSEFL11
AVERFL31=-AVSEFL31
AVERFL12= AVSEFL22
AVERFL22=AVSEFL12
AVERFL32=-AVSEFL32
AVERFL13= AVSEFL23
AVERFL23= AVSEFL13
AVERFL33=-AVSEFL33
```

WHERE: AVERFL11-33 are CIG earth to FLOLS matrix elements.
AVSEFL11-33 are standard right-handed earth to FLOLS matrix elements.

Next, the eyepoint position is converted into the FLOLS Axis Coordinate system so that the meatball angle can be computed:

NAVTRAEEQUIPCEN IH-335

$AVFLSPXP = AVXEEPFL * AVERFL11 + AVYEEPFL * AVERFL12 + AVZEEPFL * AVERFL13$
 $AVFLSPYP = AVXEEPFL * AVERFL21 + AVYEEPFL * AVERFL22 + AVZEEPFL * AVERFL23$
 $AVFLSPZP = AVXEEPFL * AVERFL31 + AVYEEPFL * AVERFL32 + AVZEEPFL * AVERFL33$

WHERE: $AVFLSPXP$, $AVFLSPYP$, $AVFLSPZP$ are X, Y, Z components of FLOLS to eyepoint vector, FLOLS Axis.
 $AVXEEPFL$, $AVYEEPFL$, $AVZEEPFL$ are X, Y, Z components of FLOLS to eyepoint vector, CIG Earth Axis.
 $AVERFL11-33$ are earth to FLOLS matrix elements.

Now the meatball angle is computed. Note in Figure C-4 that the meatball virtual image is 150. feet behind the datum line in the FLOLS Axis System (Figure C-3) and .4166 feet below the datum line:

$$AVTNFVRT = (.41666 - AVFLSZVP) / (AVFLSXVP + 150.)$$

$$AVMTBLAG = 57.29328 * AVTNFVRT$$

$$AVMEATBL = 1.2266667 * AVMTBLAG$$

WHERE: $AVTNFVRT$ is the tangent of the glideslope angle.
 $AVFLSXVP$, $AVFLSZVP$ are X, Z components of FLOLS to eyepoint.
 $AVMTBLAG$ is the meatball angle in degrees (this parameter is sent to CIG to drive FLOLS)
 $AVMEATBL$ is the drive parameter for the EOS FLOLS meatball indicator (scaled for hardware).

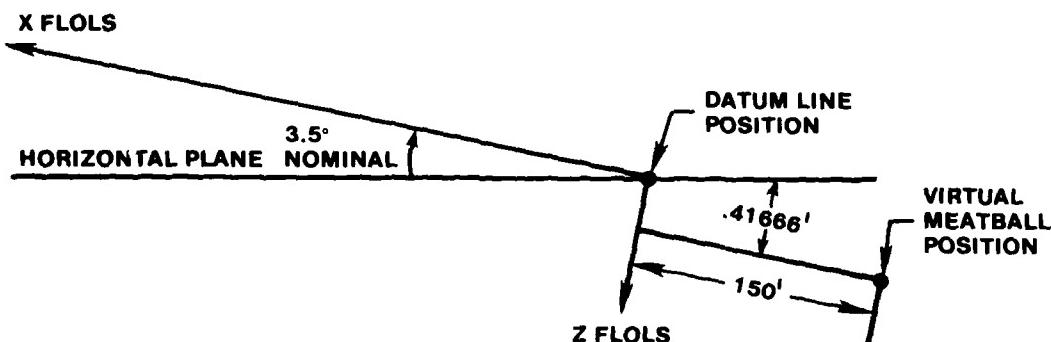


Figure C-4. FLOLS Datum Line and Meatball Position, Side View.

NAVTRAEEQUIPCEN IH-335

D. The input module, output module, and input/output parameters are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine if Carrier or
FLOLS Meridian is Loaded
- FLOLS Roll, Pitch Set at EOS
(2) WCARRIER - CVA Position, Attitude Par's

Output Module: (1) ZM6SWCGI - FLOLS X, Y, Z Position, CIG Earth,
extrapolations
- FLOLS Roll, Pitch, Yaw Angles,
extrapolations
- Meatball Angle

APPENDIX D
GENERAL PURPOSE DRIVER

This subroutine computes the drive parameters for any moving model selected by the user. This subroutine is called by the visual module ZM6SMOVE when the flag general is set true. This is a special user subroutine which allows total user control of the moving model specified. All the other drivers are bypassed in this mode.

"GENERAL" DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The "GENERAL" position and attitude are provided by the user through the experimenter operator station (EOS) or the visual initializer software. The selected target index is also set for values 1-6 to specify which model is to be driven. The user must know which moving models are available on which database to get successful results.

Pass flight position parameters to visual:

I=BJPRTGI	... Get Target Index
AVWMX(I)=AJGEXT(I)	... Pass X Position
AVWMY(I)=AJGEYT(I)	... Pass Y Position
AVWMZ(I)=AJGEZT(I)	... Pass Z Position

WHERE: AVWMX(I), AVWMY(I), AVWMZ(I) are visual X, Y, Z positions of selected target, CIG Earth Axis.

AJGEXT(I) AJGEYT(I), AJGEZT(I) are X, Y, Z flight target positions, CIG Earth Axis.

BJPRTGI is the flight target (moving model) index which should be set to values 1-6.

Pass flight attitude angles to visual parameters:

AVWMYW(I)=AJPSIT(I)	... Pass Yaw Angle
AVWMPT(I)=AJGPITCH	... Pass Pitch Angle
AVWMRL(I)=AJGROLL	... Pass Roll Angle

WHERE: AVWMRL(I), AVWMPT(I), AVWMYW(I) are visual yaw, pitch, roll "GENERAL" angles, CIG Earth Axis.
AJPSIT(I), AJGPITCH, AJGROLL are flight yaw, pitch, roll angles, CIG Earth Axis System.

B. Next, extrapolations are computed for "GENERAL" position and attitude. These are required for the CIG System which, runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

NAVTRAEEQUIPCEN IH-335

Compute the extrapolated value of "GENERAL" position 16.7 milliseconds from now:

$$\begin{aligned} \text{AVWMXD(I)} &= 1.5 * \text{AVWMX(I)} - .5 * \text{AVWMX1} \\ \text{AVWMDY(I)} &= 1.5 * \text{AVWMDY(I)} - .5 * \text{AVWMDY1} \\ \text{AVWMZD(I)} &= 1.5 * \text{AVWMZ(I)} - .5 * \text{AVWMZ1} \end{aligned}$$

WHERE: AVWMXD(I), AVWMDY(I), AVWMZD(I) are extrapolated X, Y, Z "GENERAL" position, CIG Earth Axis System.
AVWMX(I), AVWMDY(I), AVWMZ(I) are current X, Y, Z "GENERAL" position, CIG Earth Axis System.
AVWMX1, AVWMDY1, AVWMZ1 are last frame's X, Y, Z "GENERAL" position, CIG Earth Axis System.

Save the previous frame position:

$$\begin{aligned} \text{AVWMX1} &= \text{AVWMX(I)} \\ \text{AVWMDY1} &= \text{AVWMDY(I)} \\ \text{AVWMZ1} &= \text{AVWMZ(I)} \end{aligned}$$

WHERE: AVWMX1, AVWMDY1, AVWMZ1 are last frame's "GENERAL" X, Y, Z position, CIG Earth Axis System.
AVWMX(I), AVWMDY(I), AVWMZ(I) are current "GENERAL" X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of "GENERAL" attitude 16.7 milliseconds from now:

$$\begin{aligned} \text{AVWMDYWD(I)} &= 1.5 * \text{AVWMDYW(I)} - .5 * \text{AVWMDYW1} \\ \text{AVWMPTD(I)} &= 1.5 * \text{AVWMPT(I)} - .5 * \text{AVWMPT1} \\ \text{AVWMRLD(I)} &= 1.5 * \text{AVWMRL(I)} - .5 * \text{AVWMRL1} \end{aligned}$$

WHERE: AVWMDYWD(I), AVWMPTD(I), AVWMRLD(I) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
AVWMDYW(I), AVWMPT(I), AVWMRL(I) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMDYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

$$\begin{aligned} \text{AVWMDYW1} &= \text{AVWMDYW(I)} \\ \text{AVWMPT1} &= \text{AVWMPT(I)} \\ \text{AVWMRL1} &= \text{AVWMRL(I)} \end{aligned}$$

WHERE: AVWMDYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
AVWMDYW(I), AVWMPT(I), AVWMRL(I) are current yaw, pitch, roll angles, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

C. The input, output module, and the input/output parameters used in this subroutine are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request

Output Module: (2) ZM6SWCGI - "GENERAL" X, Y, Z Pos CIG Earth,
extrapolations
- "GENERAL" PT, YW, RL Ang, CIG Earth,
extrapolations

APPENDIX E
AIR-TO-SURFACE ROCKET DRIVER SUBROUTINE

This subroutine computes the drive parameters for the air-to-surface missiles (ASM's) launched from the T-2C. These ASM's are 2.75 inch rockets which are launched in pairs from launchers located on the T-2C. The release position and attitude come from flight module ZM3SA003 and position updates from module ZM3SA004. This module is called by the visual module ZM6SMOVE when weapon data bases are loaded and rockets are fired. Figure E-1 illustrates a typical rocket launch delivery.

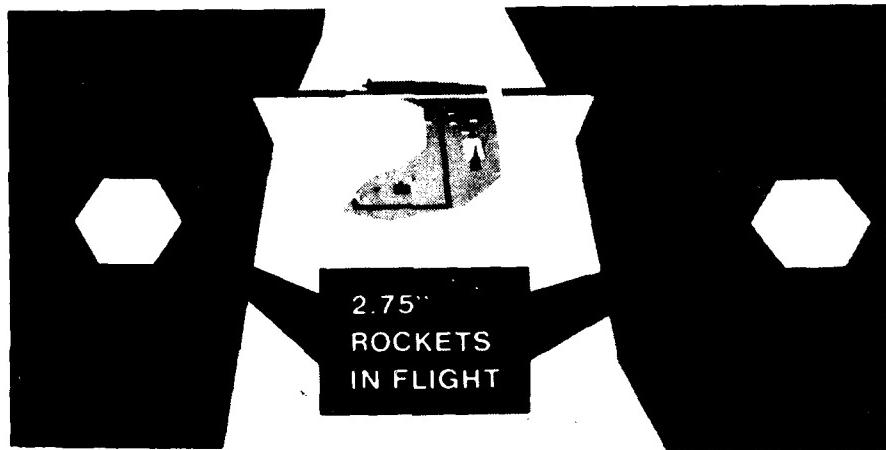


Figure E-1. Air-to-Surface Missiles (2.75" Rockets).

ROCKET DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The rocket position and attitude is defined for each of the two rockets launched in the flight software. The visual software treats the two rockets as a single moving model and needs the centroid position between the two rockets (see Figure E-2) to present the correct picture in the dome. Therefore, the two flight rocket positions must be converted into a single centroid position for the visual software. The attitude angles are the same and hence are passed directly from flight to visual.

Convert the left and right rocket positions to a single centroid value and pass to visual:

$AVWMX(2)=(AJAXPOSR+AJAXPOSL)/2.$	$+ AJBY$... Centroid Release
$AVWMY(2)=(AJAYPOSR+AJAYPOSL)/2.$	$+ AJBX$... Point + Update Value
$AVWMZ(2)=AJAZPOS-AJBZ$		

NAVTRAEEQUIPCEN IH-335

WHERE: AVWMX(2), AVWMY(2), AVWMZ(2) are visual X, Y, Z centroid position of rocket, CIG Earth Axis System.
 AJAXPOSR, AJAXPOS L are flight X position of right, left rockets, CIG Earth Axis System. (Rockets are separated by 22.42 feet on aircraft--see Figure E-2.)
 AJAYPOS R, AJAYPOS L are flight Y position of right, left rockets, CIG Earth Axis System.
 AJAZPOS is flight altitude of rockets at launch, CIG Earth Axis System.
 AJBX, AJBY, AJBZ are flight rocket X, Y, Z position update (both rockets), Flight Earth Axis System.

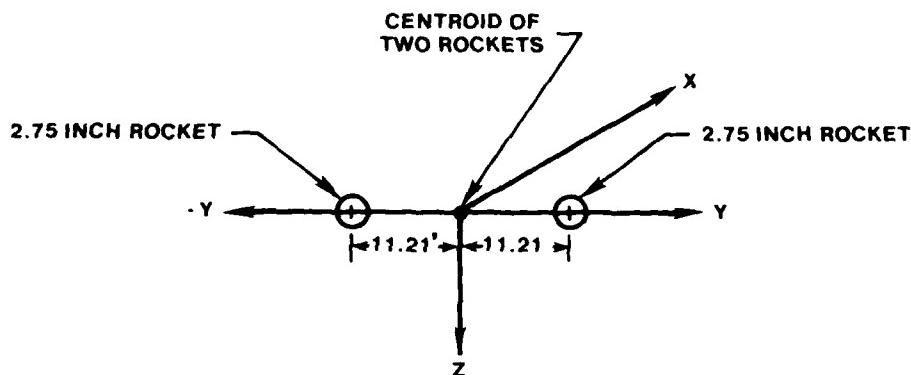


Figure E-2. Rocket Centroid Definition in Rocket Axis System.

Pass flight rocket attitude angles to visual:

AVWMYW(2)=AJBPHI
 AVWMPT(2)=AJBTHEt
 AVWMRL(2)=AJBPSI

... Pass Yaw Angle
 ... Pass Pitch Angle
 ... Pass Roll Angle

WHERE: AVWMYW(2), AVWMPT(2), AVWMRL(2) are visual yaw, pitch, roll rocket angles, CIG Earth Axis.
 AJBPSI, AJBPHI AJBPSI are flight yaw, pitch, roll rocket angles, CIG Earth Axis.

B. Next, extrapolations are computed for rocket position and attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

NAVTRAEEQUIPCEN IH-335

Compute the extrapolated value of rocket position 16.7 milliseconds from now:

$$\begin{aligned} \text{AVWMXD}(2) &= 1.5 * \text{AVWMX}(2) - .5 * \text{AVWMX1} \\ \text{AVWYD}(2) &= 1.5 * \text{AVWY}(2) - .5 * \text{AVWY1} \\ \text{AVWMZD}(2) &= 1.5 * \text{AVWMZ}(2) - .5 * \text{AVWMZ1} \end{aligned}$$

WHERE: $\text{AVWMXD}(2)$, $\text{AVWYD}(2)$, $\text{AVWMZD}(2)$ are extrapolated X, Y, Z rocket position, CIG Earth Axis System.
 $\text{AVWMX}(2)$, $\text{AVWY}(2)$, $\text{AVWMZ}(2)$ are current X, Y, Z rocket position, CIG Earth Axis System.
 AVWMX1 , AVWY1 , AVWMZ1 are last frame's X, Y, Z rocket position, CIG Earth Axis System.

Save the previous frame position:

$$\begin{aligned} \text{AVWMX1} &= \text{AVWMX}(2) \\ \text{AVWY1} &= \text{AVWY}(2) \\ \text{AVWMZ1} &= \text{AVWMZ}(2) \end{aligned}$$

WHERE: AVWMX1 , AVWY1 , AVWMZ1 are last frame's rocket X, Y, Z position, CIG Earth Axis System.
 $\text{AVWMX}(2)$, $\text{AVWY}(2)$, $\text{AVWMZ}(2)$ are current rocket X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of rocket attitude 16.7 milliseconds from now:

$$\begin{aligned} \text{AVWYWD}(2) &= 1.5 * \text{AVWYW}(2) - .5 * \text{AVWYW1} \\ \text{AVWMPTD}(2) &= 1.5 * \text{AVWMPT}(2) - .5 * \text{AVWMPT1} \\ \text{AVWMRLD}(2) &= 1.5 * \text{AVWMRL}(2) - .5 * \text{AVWMRL1} \end{aligned}$$

WHERE: $\text{AVWYWD}(2)$, $\text{AVWMPTD}(2)$, $\text{AVWMRLD}(2)$ are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
 $\text{AVWYW}(2)$, $\text{AVWMPT}(2)$, $\text{AVWMRL}(2)$ are current yaw, pitch, roll angles, CIG Earth Axis.
 AVWYW1 , AVWMPT1 , AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

$$\begin{aligned} \text{AVWYW1} &= \text{AVWYW}(2) \\ \text{AVWMPT1} &= \text{AVWMPT}(2) \\ \text{AVWMRL1} &= \text{AVWMRL}(2) \end{aligned}$$

WHERE: AVWYW1 , AVWMPT1 , AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
 $\text{AVWYW}(2)$, $\text{AVWMPT}(2)$, $\text{AVWMRL}(2)$ are current yaw, pitch, roll angles, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

C. The input module, output module, and parameters used by this subroutine are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request
(2) ZM3SA~~00~~3 - Gives ASM Pos, Att at Release
(3) ZM3SA~~00~~4 - Generates ASM Pos, Att, Rates

Output Module: (2) ZM6SWCGI - ASM X, Y, Z Pos CIG Earth,
extrapolations
- ASM PT, YW, RL Ang, CIG Earth,
extrapolations

AD-A117 141

NAVAL TRAINING EQUIPMENT CENTER ORLANDO FL
MATH MODEL DESCRIPTION FOR THE VISUAL TECHNOLOGY RESEARCH SIMUL--ETC(U)
JAN 82 E M HOLLER
NAVTRAEGUIPC-IH-335

F/8 12/1

UNCLASSIFIED

NL

202
4A
1774

END
DATE FILMED
01-02-82
RTIC

NAVTRAEEQUIPCEN IH-335

APPENDIX F

SURFACE-TO-AIR MISSILE (SAM) DRIVER SUBROUTINE

This module computes the drive parameters for the surface-to-air missile (SAM). This subroutine is called by the visual module ZM6SMOVE when the Twin Town database is loaded and the SAM launch flag LJBSAM is set true at the experimenter operator station (EOS). Figure F-1 shows a SAM launch.



Figure F-1. Surface-to-Air Missile (SAM) Launch.

SAM DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The first step in the SAM driver algorithm is to initialize the SAM launch parameters. This includes placement of the SAM on the launch pad (Figure F-2) and resetting SAM timers and hit flags. As shown in Figure F-2, the SAM is placed at the center of the launch pad and with a pitch angle of 90 degrees.

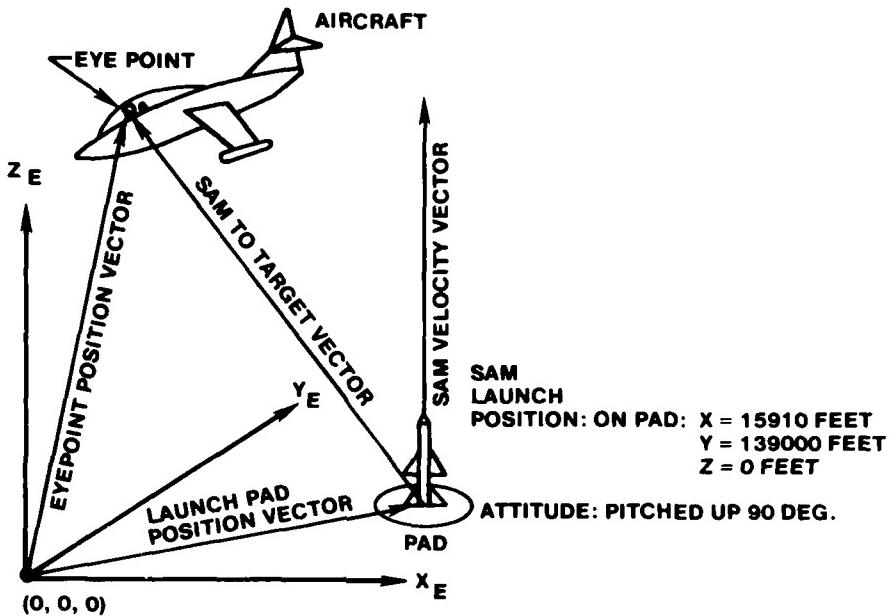


Figure F-2. SAM Launch Pad Setup, CIG Earth Axis System.

Initialize SAM starting position:

AJSAMXP=AJSAMXSP ...PAD X position coordinate = 15,910 feet
 AJSAMPY=AJSAMYSP ...PAD Y position coordinate = 139,000 feet
 AJSAMZP=AJSAMZSP ...PAD Z position coordinate = 0 feet

WHERE: AJSAMXP, AJSAMPY, AJSAMZP are SAM X, Y, Z current position, CIG Earth Axis.
 AJSAMXSP, AJSAMYSP, AJSAMZSP are X, Y, Z launch position of SAM, CIG Earth Axis System.

Initialize SAM starting attitude:

AJSAMYAW=0.
 AJSAMPTH=90.
 AJSAMROL=0.

WHERE: AJSAMYAW, AJSAMPTH, AJSAMROL are SAM attitude angles in CIG Earth Axis System.
 90 = 90 Degrees Up (SAM Pointed Straight Up)
 0 = 0 Degrees of Yaw and Roll

SAM launch flags and timers are initialized:

LJSAMFLT=.TRUE.
 LJSAMST=.FALSE.
 LJWCH =.FALSE.

NAVTRAEEQUIPCEN IH-335

LJWOH =.FALSE.
LJWSLC =.TRUE.
IJSAMHTM=0
IJSAMTIM=0

WHERE: LJSAMFLT=SAM is in Flight When = True
LJSAMSFT=Skill SAM Trajectory (Used After a Hit) When = True
LJCH=SAM Passed Within 100 Ft When = True
LJOH=SAM Passed Within 50 Ft When = True
LJWSLC=Flash Launch Pad When = True
IJSAMHTM=Sam Flight Timer, 30 Per Second
IJSAMHTM=Close/Own Hit Timer, 30 Per Second

B. The second step is to flash the SAM launch pad for 5 seconds. This is done to simulate the acquisition phase of a SAM launch. The flashing pad indicates to the pilot that the SAM site has locked on the T-2C; and is preparing to launch a SAM against him.

Let pad flash for 5 seconds, then go to SAM trajectory phase:

LJWSLC=.TRUE.	... Flash SAM Launch Pad
IJSAMTIM=IJSAMTIM+1	... Increment SAM Flight Counter
IF(IJSAMTIM.LT.150) GO TO 100	... Bypass SAM Trajectory for First 150 Loops (5 Seconds)
LJWSLC=.FALSE. (SAM Trajectory Program)	... Turn Off Launch Pad Flash ... Compute SAM Trajectory
100 CONTINUE	... Pass SAM Launch Position To Visual (Still on Pad)

C. The third step is to compute the SAM trajectory. The SAM trajectory has two phases. In the first phase, the SAM lifts off the pad vertically and gradually turns toward the aircraft, acquiring full turn capability (9 G's) after 5 seconds of flight. This phase simulates the initial SAM position and attitude alignment on the target immediately after launch. The second phase is full pursuit (9 G's) of the target until one minute of flight or pilot flies below 200 feet altitude. Refer to Figure F-3 for the trajectory description below.

Compute SAM to target vector components:

XV=AVPXCEP-AJSAMXP
YV=AVPYCEP-AJSAMYP
ZV=AVPZCEP-AJSAMZP

WHERE: XV, YV, ZV are X, Y, Z components of SAM to target vector, CIG Earth Axis System.
AVPXCEP, AVPYCEP, AVPZCEP are X, Y, Z components of eyepoint position, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

AJSAMXP, AJSAMYP, AJSAMZP are X, Y, Z components of SAM position, CIG Earth Axis System.

Compute SAM to target miss distance:

$$AJSAMSL = \sqrt{(XV^2 + YV^2 + ZV^2)}$$

WHERE: AJSAMSL is SAM to target vector magnitude in feet.
 XV, YV, ZV are X, Y, Z components of SAM to target vector, CIG Earth Axis System.

Save previous SAM velocity vector:

$$V1X = AJSAMVX$$

$$V1Y = AJSAMVY$$

$$V1Z = AJSAMVZ$$

WHERE: V1X, V1Y, V1Z are X, Y, Z components of previous SAM velocity vector, CIG Earth Axis System.
 AJSAMVX, AJSAMVY, AJSAMVZ are X, Y, Z components of current SAM velocity vector, CIG Earth Axis System.

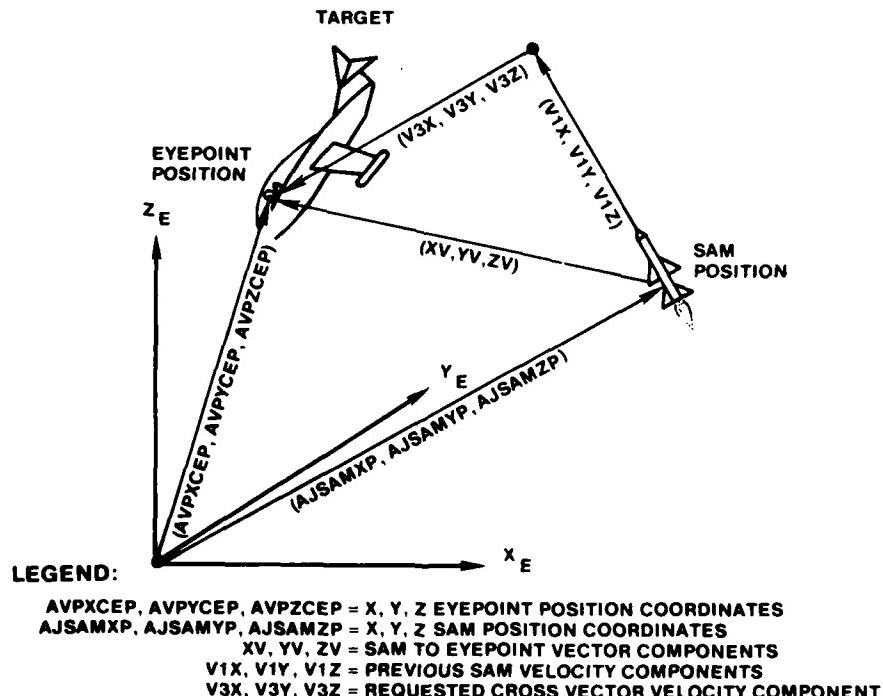


Figure F-3. SAM Trajectory Vectors, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

Compute new requested SAM velocity vector:

AJSAMVX=AJSAMVEL*XV/AJSAMSL
AJSAMVY=AJSAMVEL*YV/AJSAMSL
AJSAMVZ=AJSAMVEL*ZV/AJSAMSL

WHERE: AJSAMVX, AJSAMVY, AJSAMVZ are X, Y, Z components of SAM velocity vector, CIG Earth Axis.
AJSAMVEL is SAM velocity in ft/sec (normally 1200 ft/sec).
XV/AJSAMSL is X component of SAM to target unit vector.
YV/AJSAMSL is Y component of SAM to target unit vector.
ZV/AJSAMSL is Z component of SAM to target unit vector.

Compute requested acceleration (cross vector):

V3X=AJSAMVX-V1X
V3Y=AJSAMVY-V1Y
V3Z=AJSAMVZ-V1Z
AJSAMVC=SQRT(V3X*V3X+V3Y*V3Y+V3Z*V3Z)

WHERE: V3X, V3Y, V3Z are X, Y, Z components of requested acceleration (cross vector), CIG Earth Axis System.
AJSAMVX, AJSAMVY, AJSAMVZ are components of requested SAM velocity (to get to the target), CIG Earth Axis System.
V1X, V1Y, V1Z are X, Y, Z components of previous SAM velocity vector, CIG Earth Axis System.
AJSAMVC is magnitude of cross vector components V3X, V3Y, V3Z.

Compute maximum velocity change for specified SAM G limit:

(Maximum velocity change per pass = $32.2 * G * AQDELT2$)
AJSAMVM=32.2*AJSAMG*AQDELT2

WHERE: AJSAMVM is maximum velocity change per pass in Ft/Sec
32.2 = Acceleration of Gravity = 32.2 Ft/Sec*Sec
AJSAMG is maximum G's SAM can pull in maneuvering to target.
AQDELT2 is visual CPU integration constant (1/30 sec) per pass through this subroutine.

Let SAM turn slowly for first 5 seconds (Phase 1 Guidance):

IF(IJSAMTIM.GT.(IJSAMGT+IJSAMLT+150)) GO TO 18
AJSAMVM=(IJSAMTIM-(IJSAMLT+150))/150*AJSAMVM...INCR G'S WITH TIME
18 CONTINUE

WHERE: IJSAMTIM is SAM flight counter (increments 1 per pass).
IJSAMLT is SAM Phase 1 counter (normally set to 150), which controls Phase 1 trajectory time span.
150 = $30*5 = 5$ seconds of time for phase 1 trajectory.
AJSAMVM is maximum G's SAM can pull.

Next, the acceleration is checked to see if the maximum allowed has been exceeded. If it has, only the maximum value is used in the trajectory computation. Otherwise, the requested acceleration required to track the target is used.

Test Acceleration to see if it exceeds maximum value allowed:

IF(AJSAMVC.LT.AJSAMVM) GO TO 20 ... Bypass Rate Limit Code
 If G's Not Exceeded

WHERE: AJSAMVC is the requested velocity change (acceleration) to guide to target in Ft/Sec*Sec.
AJSAMVM is the maximum velocity change (acceleration) allowed (computed using maximum G limit).

Requested guidance will exceed maximum acceleration allowed.
Compute limited velocity vectors to guide on the target:

V3LX=V3X*AJSAMVM/AJSAMVC
V3LY=V3Y*AJSAMVM/AJSAMVC
V3LZ=V3Z*AJSAMVM/AJSAMVC

WHERE: V3LX, V3LY, V3LZ are X, Y, Z components of velocity change (acceleration limited) to track target.
V3X, V3Y, V3Z are X, Y, Z components of SAM velocity components change requested to track target.
AJSAMVM is the maximum velocity change allowed.
AJSAMVC is the requested velocity change to guide on target.

Compute new commanded SAM velocity vector (G limited):

AJSAMVX=V1X+V3LX
AJSAMVY=V1Y+V3LY
AJSAMVZ=V1Z+V3LZ

20 CONTINUE ... Branch to Here if G's Not Limited

WHERE: AJSAMVX, AJSAMVY, AJSAMVZ are new SAM X, Y, Z velocity vector components, CIG Earth Axis System.
V1X, V1Y, V1Z are X, Y, Z components of SAM previous velocity (last frame) vector, CIG Earth Axis.
V3LX, V3LY, V3LZ are X, Y, Z components of allowed velocity change to track target.

Next, the SAM position and attitude are computed. The position is computed using the velocity components determined above. The attitude is computed from the angles defined by the vector between the previous SAM position and current SAM position.

NAVTRAEEQUIPCEN IH-335

Save the previous SAM position:

AJSAMXP1=AJSAMXP
AJSAMYP1=AJSAMYP
AJSAMZP1=AJSAMZP

WHERE: AJSAMXP1, AJSAMYP1, AJSAMZP1 are X, Y, Z components of SAM position,
last frame.
AJSAMXP, AJSAMYP, AJSAMZP are X, Y, Z components of SAM current
position, CIG Earth Axis.

Compute new SAM position in CIG Earth:

AJSAMXP=AJSAMXP+AJSAMVX*AQDELT2
AJSAMYP=AJSAMYP+AJSAMVY*AQDELT2
AJSAMZP=AJSAMZP+AJSAMVZ*AQDELT2

Pass SAM position to visual (CIG):

AVWMX(1)=AJSAMXP
AVWMY(1)=AJSAMYP
AVWMZ(1)=AJSAMZP

WHERE: AJSAMXP, AJSAMYP, AJSAMZP are components of new SAM Position, CIG
Earth Axis System.
AJSAMVX, AJSAMVY, AJSAMVZ are X, Y, Z components of SAM velocity,
CIG Earth Axis System.
AQDELT2 is integration constant for visual computer.
AVWMX(1), AVWMY(1), AVWMZ(1) is the current SAM position passed to
CIG moving model 1.

Compute vector from previous SAM position to current SAM position:

AJSAMXPV=AJSAMXP-AJSAMXP1
AJSAMYPV=AJSAMYP-AJSAMYP1
AJSAMZPV=AJSAMZP-AJSAMZP1

WHERE: AJSAMXPV, AJSAMYPV, AJSAMZPV are X, Y, Z components of SAM previous
to current position vector, CIG Earth Axis System.
AJSAMXP, AJSAMYP, AJSAMZP are X, Y, Z components of SAM current
position, CIG Earth Axis.
AJSAMXP1, AJSAMYP1, AJSAMZP1 are X, Y, Z components of SAM previous
position, CIG Earth.

Compute the projection of the SAM previous to present position
vector on the X-Y CIG Earth Axis Plane. This will be used to compute the SAM
Heading angle:

AJSAMXYR=SQRT(AJSAMXPV*AJSAMXPV + AJSAMYPV*AJSAMYPV)

NAVTRAEEQUIPCEN IH-335

Compute new SAM attitude:

AJSAMPTH=ARCTAN(AJSAMZPV, AJSAMXYR)
AJSAMYAW=ARCTAN(AJSAMXPV, AJSAMPYV)
AJSAMROL=ARCTAN(AJSAMZPV, AJSAMXPV)

Pass SAM attitude to visual (CIG):

AVWMYW(1)=AJSAMYAW
AVWMPT(1)=AJSAMPTH
AVWMRL(1)=AJSAMROL

WHERE: AJSAMPTH, AJSAMYAW, AJSAMROL are yaw, pitch, and roll angles of SAM, CIG Earth Axis.
AVWMYW(1), AVWMPT(1), AVWMRL(1) is current SAM attitude passed to CIG moving model 1.
AJSAMXPV, AJSAMPYV, AJSAMZPV are X, Y, Z components of SAM previous to present position vector, CIG Earth Axis System.
AJSAMXYR is projection of SAM previous to present position vector on X-Y plane of CIG Earth Axis.

D. Next, extrapolations are computed for SAM position attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of SAM position 16.7 milliseconds from now:

AVWMXD(1)=1.5*AVWMX(1) - .5*AVWMX1
AVWMD(1)=1.5*AVWMY(1) - .5*AVWMY1
AVWMZD(1)=1.5*AVWMZ(1) - .5*AVWMZ1

WHERE: AVWMXD(1), AVWMD(1), AVWMZD(1) are extrapolated X, Y, Z SAM position, CIG Earth Axis System.
AVWMX(1), AVWMY(1), AVWMZ(1) are current X, Y, Z SAM position, CIG Earth Axis System.
AVWMX1, AVWMY1, AVWMZ1, are last frame's X, Y, Z SAM position, CIG Earth Axis System.

Save the previous frame position.

AVWMX1=AVWMX(1)
AVWMY1=AVWMY(1)
AVWMZ1=AVWMZ(1)

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frame's SAM X, Y, Z position, CIG Earth Axis System.
AVWMX(1), AVWMY(1), AVWMZ(1) are current SAM X, Y, Z position, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

Compute the extrapolated value of SAM attitude 16.7 milliseconds from now:

$$\begin{aligned} \text{AVWMYWD}(1) &= 1.5 * \text{AVWMYW}(1) - .5 * \text{AVWMYW}_1 \\ \text{AVWMPTD}(1) &= 1.5 * \text{AVWMPT}(1) - .5 * \text{AVWMPT}_1 \\ \text{AVWMRLD}(1) &= 1.5 * \text{AVWMRL}(1) - .5 * \text{AVWMRL}_1 \end{aligned}$$

WHERE: $\text{AVWMYWD}(1)$, $\text{AVWMPTD}(1)$, $\text{AVWMRLD}(1)$ are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
 $\text{AVWMYW}(1)$, $\text{AVWMPT}(1)$, $\text{AVWMRL}(1)$ are current yaw, pitch, roll angles, CIG Earth Axis.
 AVWMYW_1 , AVWMPT_1 , AVWMRL_1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

$$\begin{aligned} \text{AVWMYW}_1 &= \text{AVWMYW}(1) \\ \text{AVWMPT}_1 &= \text{AVWMPT}(1) \\ \text{AVWMRL}_1 &= \text{AVWMRL}(1) \end{aligned}$$

WHERE: AVWMYW_1 , AVWMPT_1 , AVWMRL_1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
 $\text{AVWMYW}(1)$, $\text{AVWMPT}(1)$, $\text{AVWMRL}(1)$ are current yaw, pitch, roll angles, CIG Earth Axis System.

E. The input modules, output modules, and input/output parameters used by this routine are summarized below:

Input Module: (1) ZM6SWCGI - SAM X, Y, Z Start Pos, CIG Earth
(2) ZM6SWØ33 - X, Y, Z Eyept Pos, CIG Earth
- SAM Launch, T/F

Output Module: (1) ZM6SWCGI - SAM X, Y, Z Pos, CIG Earth,
extrapolations
- SAM PT, YW, RL, CIG Earth,
extrapolations
- Launch, Close Hit, Hit, T/F

APPENDIX G

TA4J DRIVER SUBROUTINE

This subroutine computes the drive parameters for the TA4J. Position, heading, velocity commands come from flight module ZM3SN00A. The altitude is set at the EOS station. This subroutine is called by the visual module ZM6SMOVE when the flag EAFORFLT is set at the experimenter operator station (EOS). Figure G-1 shows the TA4J in a typical formation flight scene.



Figure G-1. TA4J in Formation Flight.

TA4J DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The TA4J position and attitude are defined relative to the center of gravity (C.G.) in both the flight and visual software. Therefore, no conversion is required for position and attitude values. Hence this subroutine merely passes flight position and attitude parameters directly to visual.

Pass flight position parameters to visual:

```
AVWMX(2)=ANCGEX
AVWMY(2)=ANCHEY
AVWMZ(2)=AVHEVECR
```

WHERE: AVWMX(2), AVWMY(2), AVWMZ(2) are visual X, Y, Z position of TA4J C.G., CIG Earth Axis System.
 ANCGEX, ANCHEY are flight X, Y position of TA4J C.G., in CIG Earth Axis System.
 AVHEVECR is the altitude of TA4J above sea level (set at EOS) in CIG Earth Axis System.

Pass flight attitude angles to visual parameters:

AVWMYW(2)=ANCVAT	... Pass Yaw Angle
AVWMPT(2)=ANTHEC	... Pass Pitch Angle
AVWMRL(2)=ANPHIC	... Pass Roll Angle

WHERE: AVWMRL(2), AVWMPT(2), AVWMYW(2) are visual yaw, pitch, roll TA4J angles, CIG Earth Axis.
 ANPHIC, ANTHEC, ANCVAT are flight yaw, pitch, roll TA4J angles, CIG Earth Axis.

B. Next, extrapolations are computed for TA4J position and attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of TA4J position 16.7 milliseconds from now:

```
AVWMXD(2)=1.5*AVWMX(2) - .5*AVWMX1
AVWMYD(2)=1.5*AVWMY(2) - .5*AVWMY1
AVWMZD(2)=1.5*AVWMZ(2) - .5*AVWMZ1
```

WHERE: AVWMXD(2), AVWMYD(2), AVWMZD(2) are extrapolated X, Y, Z TA4J position, CIG Earth Axis System.
 AVWMX(2), AVWMY(2), AVWMZ(2) are current X, Y, Z TA4J position, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z TA4J position, CIG Earth Axis System.

Save the previous frame position:

AVWMX1=AVWMX(2)
AVWMY1=AVWMY(2)
AVWMZ1=AVWMZ(2)

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frame's TA4J X, Y, Z position, CIG Earth Axis System.
AVWMX(2), AVWMY(2), AVWMZ(2) are current TA4J X, Y, Z position, CIG Earth Axis System.

Compute the extrapolated value of TA4J attitude 16.7 milliseconds from now:

AVWMYWD(2)=1.5*AVWMYW(2) - .5*AVWMYW1
AVWMPTD(2)=1.5*AVWMPT(2) - .5*AVWMPT1
AVWMRLD(2)=1.5*AVWMRL(2) - .5*AVWMRL1

WHERE: AVWMYWD(2), AVWMPTD(2), AVWMRLD(2) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMYW1=AVWMYW(2)
AVWMPT1=AVWMPT(2)
AVWMRL1=AVWMRL(2)

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
AVWMYW(2), AVWMPT(2), AVWMRL(2) are current yaw, pitch, roll angles, CIG Earth Axis System.

C. The input, output module, and the parameters used in this subroutine are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request
(2) ZM3SN00A - Generates Flt Input Par's

Output Module: (2) ZM6SWCGI - TA4J X, Y, Z Pos CIG Earth,
extrapolations
- TA4J PT, YW, RL Ang, CIG Earth,
extrapolations

NAVTRAEEQUIPCEN IH-335

APPENDIX H

TANK DRIVER

This subroutine computes the drive parameters for the tank. The road coordinates are determined based on the CIG database loaded and then the appropriate drive parameters are computed to make the tank move along that specified road. Figure H-1 shows the tank in a typical air-to-ground database scene.



Figure H-1. T-62 Tank.

TANK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The tank driver algorithm for the gunnery range is illustrated in Figure H-2. The tank drives back and forth on the roadway which is south of the diamond figure on the gunnery range. This roadway runs north and south between the diamond and arrow head figures on the gunnery range.

The tank position and attitude are set up for the initializer for the roadway shown in Figure H-2:

NAVTRAEEQUIPCEN IH-335

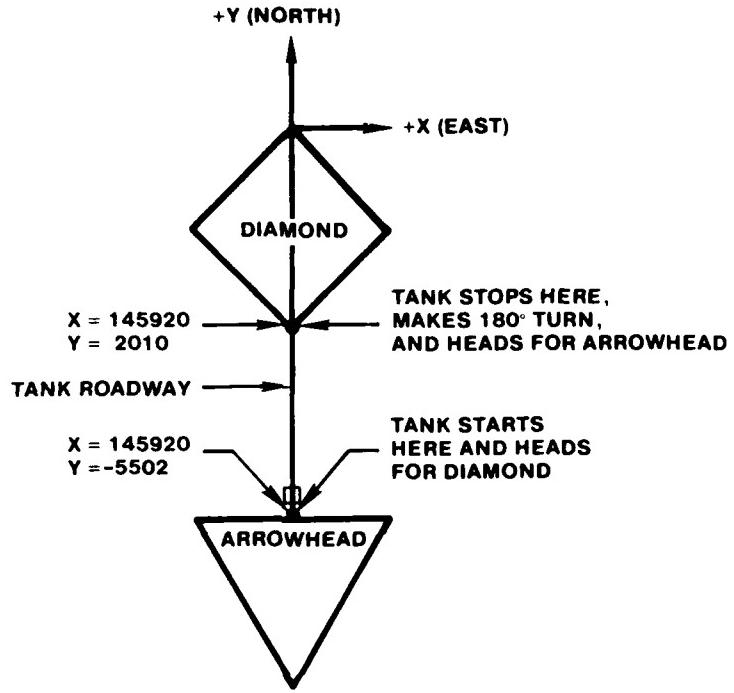


Figure H-2. Tank Route on Gunnery Range.

```

IF (IVWAES.NE.8) GO TO 90      ...Exit, if not Gunnery Range
XSTART3=145920.                .. Start Tank on South End of Roadway
YSTART3=-5502.
ZSTART3=0.0
XSTOP3=145920.                 .. Stop Tank on North End of Roadway
YSTOP3=2010.
ZSTOP3=0.0
YWSTART3=0.0                     .. Head Tank North
PTSTART3=0.
RLSTART3=0.
YWSTOP3=180.                     .. Head Tank South
PTSTOP3=0.
RLSTOP3=0.

```

WHERE: IVWAES is the visual database identification parameter (8 for Gunnery Range).
 XSTART, YSTART, ZSTART are X, Y, Z tank starting coordinates, CIG Earth Axis System.
 XSTOP, YSTOP, ZSTOP are X, Y, Z tank stopping coordinates, CIG Earth Axis System.
 YWSTART, PTSTART, RLSTART are X, Y, Z tank yaw, pitch, roll starting angles, CIG Earth Axis.
 YWSTOP, PTSTOP, RLSTOP are X, Y, Z tank yaw, pitch, roll return angles, CIG Earth Axis System.

Tank driver control flag is set up so tank is initialized at starting position only once for a given initial condition:

IF(LVTNKST) GO TO 10 .. Keep Tank Going if Already Moving
 LVTNKST=.TRUE. .. Reset Tank Control Flag For Next Pass

WHERE: LVTNKST is set true to keep driving tank.
 Is set false to restart tank at starting position.

Initialize tank starting position and attitude:
 X=XSTART3 .. Get Starting X, Y, Z Position For Tank
 Y=YSTART3
 Z=ZSTART3

ROL=RLSTART3 .. Get Starting Yaw, Pitch, Roll Angle for Tank
 PTH=PTSTART3
 YAW=YWSTART3
 10 CONTINUE .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current X, Y, Z tank position in CIG Earth Axis.

XSTART, YSTART, ZSTART are starting X, Y, Z tank position in CIG Earth Axis System.
 YAW, PTH, ROL are current tank yaw, pitch, roll angles in CIG Earth Axis System.
 RLSTART3, PTSTART3, YWSTART3 are starting yaw, pitch, roll angles in CIG Earth Axis System.

Determine if tank should be driven to the North or South:

IF((Y.LT.YSTART3).OR.(Y.LT.YSTOP3.AND.YD.GT.0.)) GO TO 20 (North)
 GO TO 30 (South)

Drive tank to the North Building.

```

20 CONTINUE          .. Entry Point to Drive Tank North
    YP=Y           .. Save Previous Position
    Y=Y+AVWTNKV*AQDELT2 .. Drive Tank to North
    YAW=YSTART3     .. Head Tank North
    YD=Y-YP         .. Compute VEL Sign
    GO TO 40        .. Output Tank Parameters to CIG

```

WHERE: YP is previous Y position, CIG Earth Axis.
Y is present Y position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is tank heading, CIG Earth Axis.
YWSTART3 is tank starting yaw angle, CIG Earth Axis.
YD is current position delta (velocity sign).

Drive tank to the South:

```

30 CONTINUE          .. Entry Point to Drive Tank South
    YP=Y           .. Save Previous Position
    Y=Y-AVWTNKV*AQDELT2 .. Drive Tank to South
    YAW=YSTOP3      .. Head Tank South
    YD=Y-YP         .. Compute VEL Sign
40 CONTINUE          .. Output Tank Parameters to CIG
    GO TO 5000

```

WHERE: YP is previous Y position, CIG Earth Axis.
Y is current Y position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current tank heading, CIG Earth Axis.
YWSTOP3 is yaw value for tank to head South.
YD is current minus previous Y position (velocity sign).

B. The tank driver algorithm for the twin town database is illustrated in Figure H-3. The tank drives around the rectangular roadway path shown in Figure H-3. The tank starts at the small town and makes a series of left hand turns at each intersection until this driver is terminated.

The tank position and attitude are set up for the initializer per the road coordinates indicated in Figure H-3:

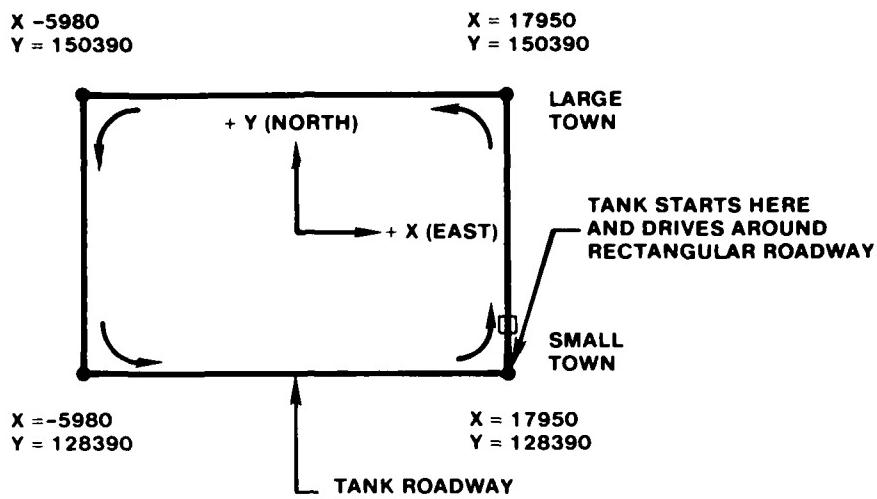


Figure H-3. Tank Route on Twin Towns.

```

90 CONTINUE          .. Entry point for twin towns
IF (IVWAES.NE.16) GO TO 190
XSTART4=17950.      .. Exit, if not twin towns
YSTART4=128390.     .. Southeast Roadway Intersection (Small Town)
ZSTART4=0.           (Tank Always Starts Here)

YWSTART4=0.0          .. Head Tank Towards Large Town (North)
PTSTART4=0.0
RLSTART4=0.0

X1=17950.            .. Northeast Roadway Intersection (Large Small)
Y1=150390.            (Tank Will Make a Left Hand Turn Here)
Z1=0.

YAW1=-90.0           .. Head Tank Left (West) to Northwest
PTH1=0.
ROL1=0.

X2=-5980.            .. Northwest Roadway Intersection
Y2=150390.
Z2=0.

YAW2=180.0           .. Head Tank South
PTH2=0.
ROL2=0.

```

NAVTRAEEQUIPCEN IH-335

X3=-5980. .. Southwest Roadway Intersection
Y3=128390.

Z3=0.0

YAW3=+90.0 .. Head Tank North to Small Town
PTH3=0.
ROL3=0.

WHERE: XSTART4, YSTART4, ZSTART4 are X, Y, Z starting coordinates, CIG Earth Axis System.
YWSTART4, PTSTART4, RLSTART4 are X, Y, Z starting yaw, pitch, roll angles, CIG Earth Axis.
X1, Y1, Z1 are X, Y, Z coordinates of Northeast intersection, (Large Town), CIG Earth Axis System.
YAW1, PTH1, ROL1 are yaw, pitch, roll angles to make tank turn left (to West), CIG Earth Axis System.
X2, Y2, Z2 are X, Y, Z coordinates of Northwest intersection, CIG Earth Axis System.
YAW2, PTH2, ROL2 are yaw, pitch, roll angles to make tank turn left (to South), CIG Earth Axis System.
X3, Y3, Z3 are X, Y, Z coordinates of Southwest intersection, CIG Earth Axis System.
YAW3, PTH3, ROL3 are yaw, pitch, roll angles to make tank turn left (to East), CIG Earth Axis System.

Tank driver control flag is set up so tank is initialized at starting position only once for a given initial condition:

IF(LVTNKST2) GO TO 100 .. Keep Tank Going if Already in Motion
LVTNKST2=.TRUE. .. Reset Tank Control Flag for Next Pass

WHERE: LVTNKST Is set to true to keep driving tank.
Is set to false to restart tank at starting position.

X=XSTART4 .. Get Starting X, Y, Z Position For Tank
Y=YSTART4
Z=ZSTART4

YAW=YWSTART4 .. Get Starting Yaw, Pitch, Roll Angle for Tank
PTH=PTSTART4
ROL=RLSTART4

100 CONTINUE .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current tank position in CIG Earth Axis System.
XSTART4, YSTART4, ZSTART4 are starting X, Y, Z tank position in CIG Earth Axis System.
YAW, PTH, ROL are current tank yaw, pitch, roll angles in CIG Earth Axis System.
YWSTART4, PTSTART4, RLSTART4 are starting yaw, pitch, roll angles in CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

Compute tank route on roadway:

```
IF(X.GE.XSTART4.AND.Y.GE.YSTART4.AND.Y.LE.Y1) GO TO 120 .. Smal Twn  
IF(Y.GE.Y1.AND.X.LE.X1.AND.X.GE.X2)      GO TO 130     .. Lg Twn  
IF(X.LE.X2.AND.Y.GT.Y3)                  GO TO 140     .. NW Inters.  
GO TO 150                                .. SW Inters.
```

Drive tank North to Northnorth intersection (large town):

```
120 CONTINUE          .. Entry Point to Drive Tank North  
Y=Y+AVWTNKV*AQDELT2 .. Drive Tank North to Large Town  
X=XSTART4           .. Put Tank on Center of Roadway  
YAW=YWSTART4        .. Head Tank North  
GO TO 160            .. Output Tank Parameters to CIG
```

WHERE: Y is present tank Y position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
X is present tank X position, CIG Earth Axis System.
AQDELT2 is visual CPU integration constant.
XSTART is tank X starting position, CIG Earth Axis.
YAW is present tank yaw, CIG Earth Axis.
YWSTART is tank yaw starting angle, CIG Earth Axis.

Drive tank South to Northwest intersection:

```
130 CONTINUE          .. Entry Point to Drive Tank South  
X=X-AVWTNKV*AQDELT2 .. Drive Tank South  
Y=Y1                 .. Put Tank on Center of Roadway  
YAW=YAW1              .. Head Tank South  
GO TO 160              .. Output Tank Parameters to CIG
```

WHERE: X is current tank X position, CIG Earth Axis System.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
Y is current tank Y position, CIG Earth Axis System.
Y1 is center of roadway between small town and large town.
YAW is current tank yaw angle, CIG Earth Axis System.
YAW1 is -90 degrees (West).

Drive tank South to Southwest intersection:

```
140 CONTINUE          .. Entry Point to Drive Tank South  
Y=Y-AVWTNKV*AQDELT2 .. Drive Tank South  
X=X2                 .. Put Tank on Center of Roadway  
YAW=YAW2              .. Head Tank South  
GO TO 160              .. Output Tank Parameters to CIG
```

WHERE: Y is current tank Y position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.

X is current tank X position, CIG Earth Axis.
 X2 is center of roadway between NW and SW corners.
 YAW is current tank yaw angle, CIG Earth Axis.
 YAW2 is 180 degrees (South heading).

Drive tank North to small town (SE corner):

150 CONTINUE	.. Entry Point to Drive Tank North
X=X+AVWTNKV*AQDELT2	.. Drive Tank North
Y=Y3	.. Keep Y on Road
YAW=YAW3	.. Head Tank North
160 CONTINUE	.. Output Tank Parameters to CIG
GO TO 5000	

WHERE: X is current tank X position, CIG Earth Axis.
 AVWTNKV is tank velocity, Ft/33 milliseconds.
 AQDELT2 is visual CPU integration constant.
 Y is current tank Y position, CIG Earth Axis.
 Y3 is center of roadway between SW corner and small town.
 YAW is current tank heading, CIG Earth Axis System.
 YAW3 is 90 degrees (East heading).

C. The tank driver algorithm for the river valley database is illustrated in Figure H-4. The tank drives back and forth on the South bridge, beginning on the West side of the bridge. This bridge is 400 feet above sea level (and the river) and is shown in Figure H-4.

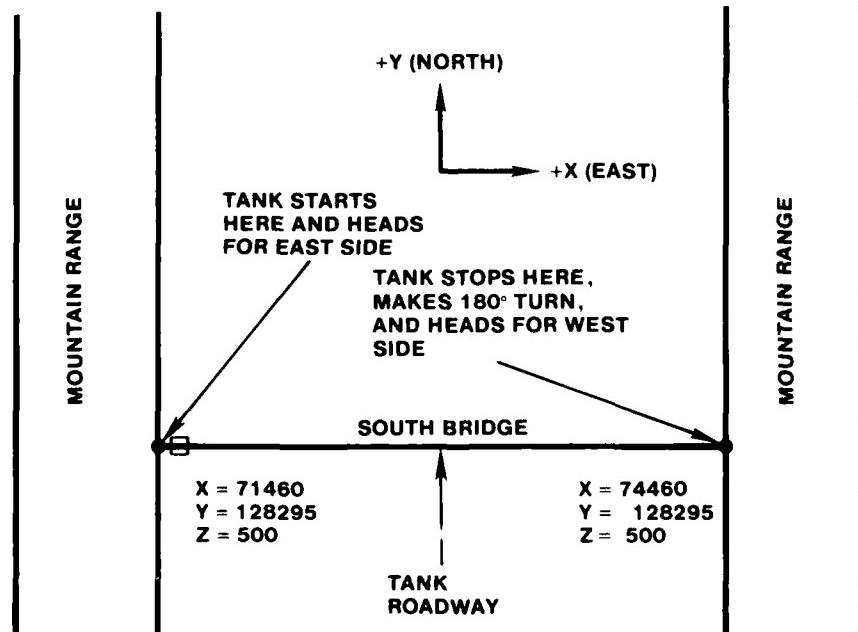


Figure H-4. Tank Route in River Valley.

NAVTRAEEQUIPCEN IH-335

The tank position and attitude are set up for the initializer per the road coordinates indicated in Figure H-4:

190 CONTINUE .. Entry point for River Valley
XSTART5=71460. .. Start Tank on West Side of Bridge
YSTART5=128295.
ZSTART5=500.

YWSTART5=+90.0 .. Head Tank East to East Side of Bridge
PTSTART5=0.
RLSTART5=0.

XSTOP5=74460. .. Stop Tank on East Side of Bridge
YSTOP5=128295.
ZSTOP5=500.

YWSTOP5=-90.0 .. Head Tank West to West Side of Bridge
PTSTOP5=0.
RLSTOP5=0.

Tank driver control flag is set up so tank is initialized at starting position only once for a given initial condition:

IF(LVTNKST3) GO TO 210 .. Keep Going if Tank Already Moving
LVTNKST3=.TRUE. .. Reset Tank Control Flag For Next Pass

WHERE: LVTNKST is set true to keep driving tank.
Is set false to restart tank at starting position.

Initialize tank starting point and attitude:

X=XSTART5 .. Get Starting X, Y, Z Position for Tank
Y=YSTART5
Z=ZSTART5

YAW=YWSTART5 .. Get Starting Yaw, Pitch, Roll Angle For
PTH=PTSTART5 Tank
ROL=RLSTART5
GO TO 220 .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current tank X, Y, Z position in CIG Earth Axis.
XSTART5, YSTART5, ZSTART5 X, Y, Z position on South side of bridge,
CIG Earth Axis System.
YAW, PTH, ROL are current tank yaw, pitch, roll angles in CIG Earth
Axis System.
YWSTART5, PTSTART5, RLSTART5 are tank yaw, pitch, roll angles, CIG
Earth Axis.

NAVTRAEEQUIPCEN IH-335

Determine if tank should be driven East or West:

IF((X.LT.XSTART5).OR.(X.LT.XSTOP5.AND.XD.GT.0.)) GO TO 220 (East)
GO TO 230 (West)

Drive tank East to East side of bridge:

220 CONTINUE	.. Entry Point to Drive Tank East
XP=X	.. Save Previous Position
X=X+AVWTNKV*AQDELT2	.. Drive Tank East
YAW=YWSTART5	.. Head Tank East
XD=X-XP	.. Compute VEL Sign
GO TO 5000	.. Output Tank Parameters to CIG

WHERE: XP is previous tank X position, CIG Earth Axis.
X is current tank X position, CIG Earth Axis.
AVWTNKV is tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current tank heading, CIG Earth Axis System.
YWSTART5 is 90 degrees (heading East).
XD is current position delta (velocity sign).

Drive Tank West to West Side of Bridge:

230 CONTINUE	.. Entry Point to Drive Tank West
XP=X	.. Save Previous Position
X=X-AVWTNKV*AQDELT2	.. Drive Tank West
YAW=YWSTOP5	.. Head Tank West
XD=X-XP	.. Compute VEL Sign

WHERE: XP is previous tank X position, CIG Earth Axis.
X is current tank X position, CIG Earth Axis.
AVWTNKV is current tank velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current tank heading, CIG Earth Axis System.
YWSTOP5 is -90 degrees (West heading).
XD is current position delta (velocity sign).

5000 CONTINUE .. Entry point to CIG data output

D. Next, extrapolations are computed for tank position attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

First, pass current tank position to visual (CIG):

AVWMX(3)=X
AVWMY(3)=Y
AVWMZ(3)=Z

NAVTRAEEQUIPCEN IH-335

Compute the extrapolated value of tank position 16.7 milliseconds from now:

$$\begin{aligned}AVWMXD(3) &= 1.5 * AVWMX(3) - .5 * AVWMX1 \\AVWEMYD(3) &= 1.5 * AVWEMY(3) - .5 * AVWEMY1 \\AVWMZD(3) &= 1.5 * AVWMZ(3) - .5 * AVWMZ1\end{aligned}$$

WHERE: $AVWMXD(3)$, $AVWEMYD(3)$, $AVWMZD(3)$ are extrapolated X, Y, Z tank position, CIG Earth Axis System.
 $AVWMX(3)$, $AVWEMY(3)$, $AVWMZ(3)$ are current X, Y, Z tank position, CIG Earth Axis System.
 $AVWMX1$, $AVWEMY1$, $AVWMZ1$ are last frame's X, Y, Z tank position, CIG Earth Axis System.

Save the previous frame position

$$\begin{aligned}AVWMX1 &= AVWMX(3) \\AVWEMY1 &= AVWEMY(3) \\AVWMZ1 &= AVWMZ(3)\end{aligned}$$

WHERE: $AVWMX1$, $AVWEMY1$, $AVWMZ1$ are last frame's tank X, Y, Z position, CIG Earth Axis System.
 $AVWMX(3)$, $AVWEMY(3)$, $AVWMZ(3)$ are current tank X, Y, Z position, CIG Earth Axis System.

First, pass current tank attitude to visual (CIG):

$$\begin{aligned}AVWEMYW(3) &= YAW \\AVWMPT(3) &= PTH \\AVWMRL(3) &= ROL\end{aligned}$$

Compute the extrapolated value of tank attitude 16.7 milliseconds from now:

$$\begin{aligned}AVWEMYWD(3) &= 1.5 * AVWEMYW(3) - .5 * AVWEMYW1 \\AVWMPTD(3) &= 1.5 * AVWMPT(3) - .5 * AVWMPT1 \\AVWMRLD(3) &= 1.5 * AVWMRL(3) - .5 * AVWMRL1\end{aligned}$$

WHERE: $AVWEMYWD(3)$, $AVWMPTD(3)$, $AVWMRLD(3)$ are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
 $AVWEMYW(3)$, $AVWMPT(3)$, $AVWMRL(3)$ are current yaw, pitch, roll angles, CIG Earth Axis.
 $AVWEMYW1$, $AVWMPT1$, $AVWMRL1$ are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

$$\begin{aligned}AVWEMYW1 &= AVWEMYW(3) \\AVWMPT1 &= AVWMPT(3) \\AVWMRL1 &= AVWMRL(3)\end{aligned}$$

NAVTRAEEQUIPCEN IH-335

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis System.
AVWMYW(3), AVWMPT(3), AVWMRL(3) are current yaw, pitch, roll angles, CIG Earth Axis System.

E. The input module, output module, and input/output parameters are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request.
- Sets Gun Range, Twin Town, River Valley Flags
- Resets Tank Start Point Flag

Output Module: (3) ZM6SWCGI - Tank X, Y, Z Pos CIG Earth, Extrapolations
- Tank Pt, Yw, R1 Ang, CIG Earth, Extrapolations

NAVTRAEEQUIPCEN IH-335

APPENDIX I
TRUCK DRIVER

This subroutine computes the drive parameters for the truck convoy. The road coordinates are determined based on the CIG database loaded and then the appropriate drive parameters are computed to make the trucks move along that specified road. Figure I-1 shows the truck convoy in a typical air-to-ground database scene.

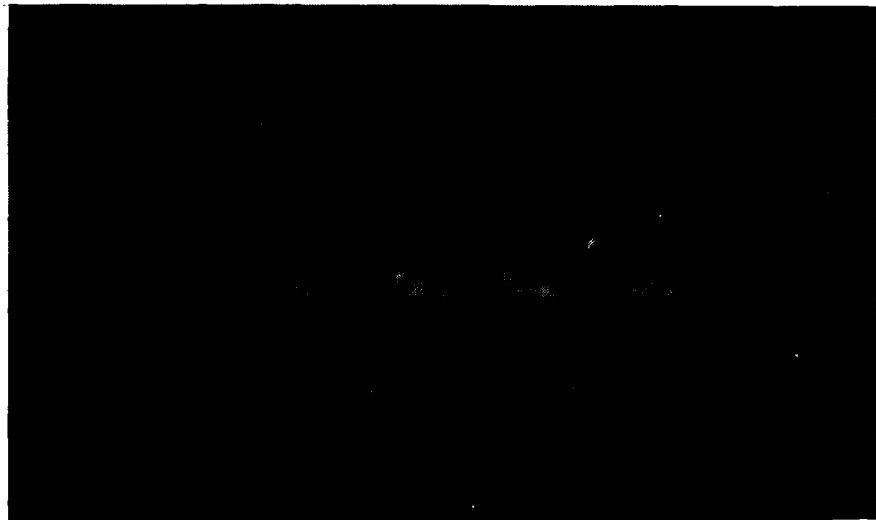


Figure I-1. Truck Convoy.

TRUCK DRIVER SUBROUTINE FUNCTIONAL DESCRIPTION

A. The truck driver algorithm for the gunnery range is illustrated in Figure I-2. The trucks drive back and forth on the roadway which is north of the gunnery range. This road runs east and west between the two points indicated in Figure I-2.

NAVTRAEEQUIPCEN IH-335

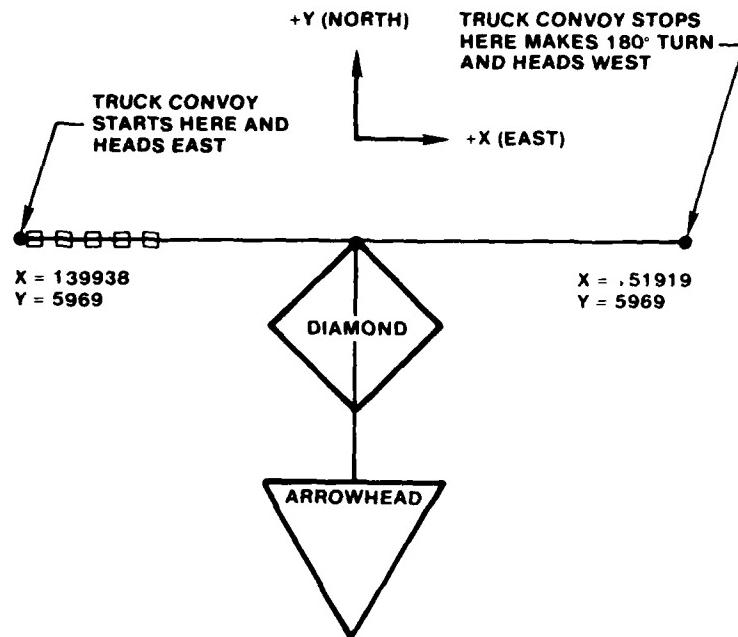


Figure I-2. Truck Convoy Route on Gunnery Range.

The truck position and attitude are set up for the initializer for the roadway shown in Figure I-2:

```

IF (IVWAES.NE.8) GO TO 90      .. Exit, if not Gunnery Range
XSTART3=139938.                .. Start Truck on West End of Roadway
YSTART3=5969.
ZSTART3=0.0

XSTOP3=.51919.                 .. Stop Truck on East End of Roadway
YSTOP3=5969.
ZSTOP3=0.0

YWSTART3=90.0                  .. Head Truck East
PTSTART3=0.
RLSTART3=0.

YWSTOP3=-90.                   .. Head Truck West
PTSTOP3=0.
RLSTOP3=0.

```

WHERE: XSTART, YSTART, ZSTART are X, Y, Z truck starting coordinates, CIG Earth Axis System.
 XSTOP, YSTOP, ZSTOP are X, Y, Z truck stopping coordinates, CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

YWSTART, PTSTART, RLSTART are X, Y, Z truck yaw, pitch, roll starting angles, CIG Earth Axis.
YWSTOP, PTSTOP, RLSTOP are X, Y, Z truck yaw, pitch, roll return angles, CIG Earth Axis System.

Truck driver control flag is set up so truck is initialized at starting position only once for a given initial condition:

IF(LVTRKST) GO TO 10 .. Keep Truck Going if Already Moving
LVTRKST=.TRUE. .. Reset Truck Control Flag For Next Pass

WHERE: LVTRKST is set true to keep driving truck.
Is set false to restart truck at starting position.

Initialize truck starting position and attitude:

X=XSTART3 .. Get Starting X, Y, Z Position For Truck
Y=YSTART3
Z=ZSTART3

ROL=RLSTART3 .. Get Starting Yaw, Pitch, Roll Angle For
PTH=PTSTART3
YAW=YWSTART3

10 CONTINUE .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current X, Y, Z truck position in CIG Earth Axis.
XSTART, YSTART, ZSTART are starting X, Y, Z truck position in CIG Earth Axis System.
YAW, PTH, ROL are current truck yaw, pitch, roll angles in CIG Earth Axis System.
RLSTART3, PTSTART3, YWSTART3 are starting yaw, pitch, roll angles in CIG Earth Axis System.

Determine if truck should be driven to the East or West:

IF((X.LT.XSTART3).OR.(X.LT.XSTOP3.AND.XD.GT.0.)) GO TO 20 (East)
GO TO 30 (West)

Drive truck to the East Building.

20 CONTINUE .. Entry Point to Drive Truck East
XP=X .. Save Previous Position
X=X+AVWTRKV*AQDELT2 .. Drive Truck to East
YAW=YWSTART3 .. Head Truck East
XD=X-XP .. Compute VEL Sign
GO TO 40 .. Output Truck Parameters to CIG

NAVTRAEEQUIPCEN IH-335

WHERE XP is previous X position, CIG Earth Axis.
 X is present X position, CIG Earth Axis.
 AVWTRKV is truck velocity, Ft/33 milliseconds.
 AQDELT2 is visual CPU integration constant.
 YAW is truck heading, CIG Earth Axis.
 YWSTART3 is truck starting yaw angle, CIG Earth Axis.
 XD is current position delta (velocity sign).

Drive truck to the West Building.

30 CONTINUE	.. Entry Point to Drive Truck West
XP=X	.. Save Previous Position
X=X-AVWTRKV*AQDELT2	.. Drive Truck to West
YAW=YWSTOP3	.. Head Truck West
XD=X-XP	.. Compute VEL Sign
40 CONTINUE	.. Output Truck Parameters to CIG
GO TO 240	

WHERE: XP is previous X position, CIG Earth Axis.
 X is current X position, CIG Earth Axis.
 AVWTRKV is truck velocity, Ft/33 milliseconds.
 AQDELT2 is visual CPU integration constant.
 YAW is current truck heading, CIG Earth Axis.
 YWSTOP3 is yaw value for truck to head West.
 XD is current minus previous X position (velocity sign).

B. The truck driver algorithm for the twin town database is illustrated in Figure I-3. The truck drives around the rectangular roadway path shown in Figure I-3. The truck starts at the small town and makes a series of left hand turns at each intersection until this driver is terminated.

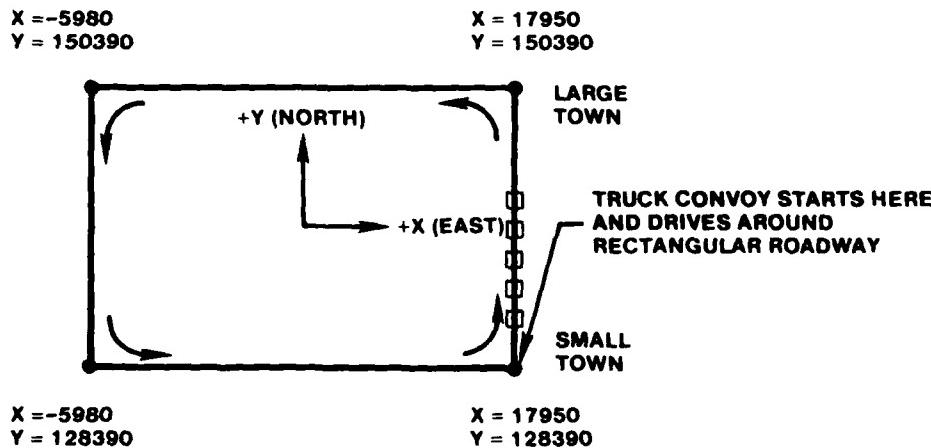


Figure I-3. Truck Convoy Route in Twin Towns.

The truck position and attitude are set up for the initializer per the road coordinates indicated in Figure I-3:

```

90 CONTINUE          .. Entry point for Twin Towns
    IF (IVWAES.NE.16) GO TO 190
                    .. Exit, if not Twin Towns
    XSTART4=17950      .. Southeast Roadway Intersection (Small Town)
    YSTART4=128390      (Truck Always Starts Here)
    ZSTART4=0.

    YWSTART4=0.0        .. Head Truck Towards Large Town (North)
    PTSTART4=0.0
    RLSTART4=0.0

    X1=17950.          .. Northeast Roadway Intersection (Large Small)
    Y1=150390          (Truck Will Make a Left Hand Turn Here)
    Z1=0.

    YAW1=-90.0         .. Head Truck Left (West) to Northwest
    PTH1=0.             Roadway Intersection
    ROL1=0.

    X2=-5980.          .. Northwest Roadway Intersection
    Y2=150390.
    Z2=0.

    YAW2=180.0         .. Head Truck South
    PTH2=0.
    ROL2=0.

    X3=-5980.          .. Southwest Roadway Intersection
    Y3=128390.
    Z3=0.0

    YAW3=+90.0         .. Head Truck East to Small Town
    PTH3=0.
    ROL3=0.

```

WHERE: XSTART4, YSTART4, ZSTART4 are X, Y, Z starting coordinates, CIG Earth Axis System.
 YWSTART4, PTSTART4, RLSTART4 are X, Y, Z starting yaw, pitch, roll angles, CIG Earth Axis.
 X1, Y1, Z1 are X, Y, Z coordinates of Northeast intersection, (Large Town), CIG Earth Axis System.
 YAW1, PTH1, ROL1 are yaw, pitch, roll angles to make truck turn left (to West), CIG Earth Axis System.
 X2, Y2, Z2 are X, Y, Z coordinates of Northwest intersection, CIG Earth Axis System.
 YAW2, PTH2, ROL2 are yaw, pitch, roll angles to make truck turn left (to South), CIG Earth Axis System.

NAVTRAEEQUIPCEN IH-335

X3, Y3, Z3 are X, Y, Z coordinates of Southwest intersection, CIG Earth Axis System.
YAW, PTH3, ROL3 are yaw, pitch, roll angles to make truck turn left (to East), CIG Earth Axis System.

Truck driver control flag is set up so truck is initialized at starting position only once for a given initial condition:

IF(LVTRKST2) GO TO 100 .. Keep Truck Going if Already in Motion
LVTRKST2=.TRUE. .. Reset Truck Control Flag For Next Pass

WHERE: LVTRKST is set to true to keep driving truck.
Is set to false to restart truck at starting position.

X=XSTART4 .. Get Starting X, Y, Z Position For Truck
Y=YSTART4
Z=ZSTART4

YAW=YWSTART4 .. Get Starting Yaw, Pitch, Roll Angle For Truck
PTH=PTSTART4
ROL=RLSTART4
100 CONTINUE .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current truck position in CIG Earth Axis System.
XSTART4, YSTART4, ZSTART4 are starting X, Y, Z truck position in CIG Earth Axis System.
YAW, PTH, ROL are current truck yaw, pitch, roll angles in CIG Earth Axis System.
YWSTART4, PTSTART4, RLSTART4 are starting yaw, pitch, roll angles in CIG Earth Axis System.

Compute truck route on roadway:

IF(X.GE.XSTART4.AND.Y.GE.YSTART4.AND.Y.LE.Y1) GO TO 120 .. Smal Twn
IF(Y.GE.Y1.AND.X.LE.X1.AND.X.GE.X2) GO TO 130 .. Lg Twn
IF(X.LE.X2.AND.Y.GT.Y3) GO TO 140 .. NW Inters
GO TO 150 .. SW Inters

Drive truck North to Northeast intersection (large town):

120 CONTINUE .. Entry Point to Drive Truck North
Y=Y+AVWTRKV*AQDELT2 .. Drive Truck North to Large Town
X=XSTART4 .. Put Truck on Center of Roadway
YAW=YWSTART4 .. Head Truck North
GO TO 160 .. Output Truck Parameters to CIG

WHERE: Y is present truck Y position, CIG Earth Axis.
AVWTRKV is truck velocity, Ft/33 milliseconds.
X is present truck X position, CIG Earth Axis System.
AQDELT2 is visual CPU integration constant.

NAVTRAEOUIPCEN IH-335

XSTART is truck X starting position, CIG Earth Axis.
YAW is present truck yaw, CIG Earth Axis.
YWSTART is truck yaw starting angle, CIG Earth Axis.

Drive truck West to Northwest intersection:

130 CONTINUE .. Entry Point to Drive Truck West
X=X-AVWTRKV*AQDELT2 .. Drive Truck West
Y=Y1 .. Put Truck on Center of Roadway
YAW=YAW1 .. Head Truck West
GO TO 160 .. Output Truck Parameters to CIG

WHERE: X is current truck X position, CIG Earth Axis System.
AVWTRKV is truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
Y is current truck Y position, CIG Earth Axis System.
Y1 is center of roadway between small town and large town.
YAW is current truck yaw angle, CIG Earth Axis System.
YAW1 is -90 degrees (West).

Drive truck South to Southwest intersection:

140 CONTINUE .. Entry Point to Drive Truck South
Y=Y-AVWTRKV*AQDELT2 .. Drive Truck South
X=X2 .. Put Truck on Center of Roadway
YAW=YAW2 .. Head Truck South
GO TO 160 .. Output Truck Parameters to CIG

WHERE: Y is current truck Y position, CIG Earth Axis.
AVWTRKV is truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
X is current truck X position, CIG Earth Axis.
X2 is center of roadway between NW and SW corners.
YAW is current truck yaw angle, CIG Earth Axis.
YAW2 is 180 degrees (South heading).

Drive truck East to small town (SE corner):

150 CONTINUE .. Entry Point to Drive Truck East
X=X+AVWTRKV*AQDELT2 .. Drive Truck East
Y=Y3 .. Keep Y on Road
YAW=YAW3 .. Head Truck East
160 CONTINUE .. Output Truck Parameters to CIG
GO TO 240

WHERE: X is current truck X position, CIG Earth Axis.
AVWTRKV is truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
Y is current truck Y position, CIG Earth Axis.

Y3 is center of roadway between SW corner and small town.
 YAW is current truck heading, CIG Earth Axis System.
 YAW3 is 90 degrees (East heading).

WHERE: X is current truck X position, CIG Earth Axis.
 AVWTRKV is truck velocity, Ft/33 milliseconds.
 AQDELT2 is visual CPU integration constant.
 Y is current truck Y position, CIG Earth Axis.
 Y3 is center of roadway between SW corner and small town.
 YAW is current truck heading, CIG Earth Axis System.
 YAW3 is 90 degrees (East heading).

C. The truck driver algorithm for the river valley database is illustrated in Figure I-4. The truck drives back and forth on the North bridge, beginning on the West side of the bridge. This bridge is 400 feet above sea level (and the river) and is shown in Figure I-4.

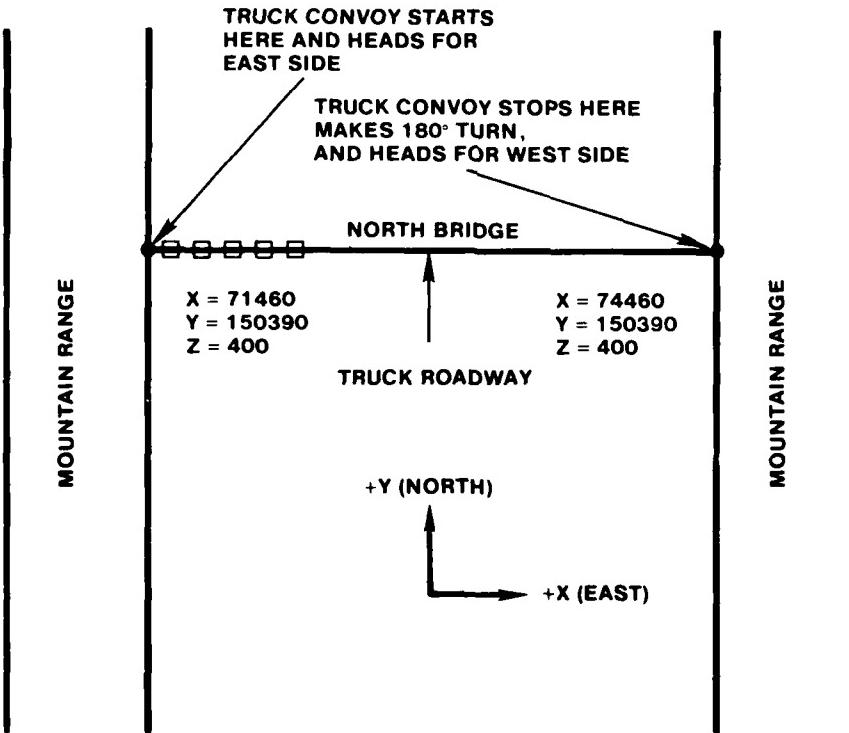


Figure I-4. Truck Convoy in River Valley.

The truck position and attitude are set up for the initializer per the road coordinates indicated in Figure I-4:

NAVTRAEEQUIPCEN IH-335

190 CONTINUE .. Entry point for River Valley
XSTART5=71460. .. Start Truck on West Side of Bridge
YSTART5=150390.
ZSTART5=400.

YWSTART5=+90.0 .. Head Truck East to East Side of Bridge
PTSTART5=0.
RLSTART5=0.

XSTOP5=74460. .. Stop Truck on East Side of Bridge
YSTOP5=150390.
ZSTOP5=400.

YWSTOP5=-90.0 .. Head Truck West to West Side of Bridge
PTSTOP5=0.
RLSTOP5=0.

Truck driver control flag is set up so truck is initialized at starting position only once for a given initial condition:

IF(LVTRKST3) GO TO 210 .. Keep Going if Truck Already Moving
LVTRKST3=.TRUE. .. Reset Truck Control Flag For Next Pass

WHERE: LVTRKST is set true to keep driving truck.
Is set false to restart truck at starting position.

Initialize truck starting point and attitude:

X=XSTART5 .. Get Starting X, Y, Z Position For Truck
Y=YSTART5
Z=ZSTART5

YAW=YWSTART5 .. Get Starting Yaw, Pitch, Roll Angle For
PTH=PTSTART5 Truck
ROL=RLSTART5
GO TO 220 .. Entry Point if Initialization Bypassed

WHERE: X, Y, Z are current truck X, Y, Z position in CIG Earth Axis.
XSTART5, YSTART5, ZSTART5 X, Y, Z position on West side bridge, CIG
Earth Axis System.
YAW, PTH, ROL are current truck yaw, pitch, roll angles in CIG Earth
Axis System.
YWSTART5, PTSTART5, RLSTART5 are truck yaw, pitch, roll angles, CIG
Earth Axis.

Determine if truck should be driven East or West:

IF((X.LT.XSTART5).OR.(X.LT.XSTOP5.AND.XD.GT.0.)) GO TO 220 (East)
GO TO 230 (West)

NAVTRAEEQUIPCEN IH-335

Drive truck East to East side of bridge:

220 CONTINUE	.. Entry Point to Drive Truck East
XP=X	.. Save Previous Position
X=X+AVWTRKV*AQDELT2	.. Drive Truck East
YAW=YSTART5	.. Head Truck East
XD=X-XP	.. Compute VEL Sign
GO TO 240	.. Output Truck Parameters to CIG

WHERE: XP is previous truck X position, CIG Earth Axis.
X is current truck X position, CIG Earth Axis.
AVWTRKV is truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current truck heading, CIG Earth Axis System.
YSTART5 is 90 degrees (heading East).
XD is current position delta (velocity sign).

Drive truck West to West side of bridge:

230 CONTINUE	.. Entry Point to Drive Truck West
XP=X	.. Save Previous Position
X=X-AVWTRKV*AQDELT2	.. Drive Truck West
YAW=YSTOP5	.. Head Truck West
XD=X-XP	.. Compute VEL Sign
240 CONTINUE	.. Entry point to CIG data output

WHERE: XP is previous truck X position, CIG Earth Axis.
X is current truck X position, CIG Earth Axis.
AVWTRKV is current truck velocity, Ft/33 milliseconds.
AQDELT2 is visual CPU integration constant.
YAW is current truck heading, CIG Earth Axis System.
YSTOP5 is -90 degrees (West heading).
XD is current position delta (velocity sign).

D. Next, extrapolations are computed for truck position attitude. These are required for the CIG System which runs at 60 Hertz rate, while the weapon visual software runs at 30 Hertz. This extrapolation is derived in Section II. The same technique is used for both position and attitude.

Compute the extrapolated value of truck position 16.7 milliseconds from now:

First, pass truck position to visual (CIG):

AVWMX(4)=X
AVWMY(4)=Y
AVWMZ(4)=Z

NAVTRAEEQUIPCEN IH-335

AVWMXD(4)=1.5*AVWMX(4) - .5*AVWMX1
AVWMYD(4)=1.5*AVWMY(4) - .5*AVWMY1
AVWMZD(4)=1.5*AVWMZ(4) - .5*AVWMZ1

WHERE: AVWMXD(4), AVWMYD(4), AVWMZD(4) are extrapolated X, Y, Z truck position, CIG Earth Axis System.
AVWMX(4), AVWMY(4), AVWMZ(4) are current X, Y, Z truck position, CIG Earth Axis System.
AVWMX1, AVWMY1, AVWMZ1 are last frame's X, Y, Z truck position, CIG Earth Axis System.

Save the previous frame position.

AVWMX1=AVWMX(4)
AVWMY1=AVWMY(4)
AVWMZ1=AVWMZ(4)

WHERE: AVWMX1, AVWMY1, AVWMZ1 are last frame's truck X, Y, Z position, CIG Earth Axis System.
AVWMX(4), AVWMY(4), AVWMZ(4) are current truck X, Y, Z position, CIG Earth Axis System.

First, pass truck attitudes to visual (CIG):

AVWMYW(4)=YAW
AVWMPT(4)=PTH
AVWMRL(4)=ROL

Compute the extrapolated value of truck attitude 16.7 milliseconds from now:

AVWMYWD(4)=1.5*AVWMYW(4) - .5*AVWMYW1
AVWMPTD(4)=1.5*AVWMPT(4) - .5*AVWMPT1
AVWMRLD(4)=1.5*AVWMRL(4) - .5*AVWMRL1

WHERE: AVWMYWD(4), AVWMPTD(4), AVWMRLD(4) are extrapolated yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW(4), AVWMPT(4), AVWMRL(4) are current yaw, pitch, roll angles, CIG Earth Axis.
AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles, CIG Earth Axis.

Save last frame attitude values:

AVWMYW1=AVWMYW(4)
AVWMPT1=AVWMPT(4)
AVWMRL1=AVWMRL(4)

NAVTRAEEQUIPCEN IH-335

WHERE: AVWMYW1, AVWMPT1, AVWMRL1 are last frame's yaw, pitch, roll angles,
CIG Earth Axis System.
AVWMYW(4), AVWMPT(4), AVWMRL(4) are current yaw, pitch, roll angles,
CIG Earth Axis System.

E. The input module, output module, and input/output parameters are listed below:

Input Module: (1) ZM6SMOVE - Calls This Routine Upon Request
- Sets Gun Range, Twin Town, River
Valley Flags
- Resets Truck Start Pt Flag

Output Module: (4) ZM6SWCGI - Truck X, Y, Z Pos CIG Earth,
Extrapolations
- Truck PT, YW, RL Ang, CIG Earth,
Extrapolations

NAVTRAEEQUIPCEN IH-335

Distribution List

Naval Training Equipment Center Orlando, FL 32813	60	Dr. Jesse Orlansky Science and Technology Division Institute for Defense Analyses	1
Commander HQ, TRADOC Attn: ATTNG-PA Ft. Monroe, VA 23651	1	400 Army-Navy Drive Arlington, VA 22202	
Center Library Naval Personnel Research and Development Center San Diego, CA 92152		Chief of Naval Operations OP-987H	1
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2	Attn: Dr. R.G. Smith Washington, D.C. 20350	
Library Division of Public Documents Government Printing Office Washington, D.C. 20402		Scientific Technical Information Office	1
Director Training Analysis & Evaluation Group Department of the Navy Orlando, FL 32813	12	NASA Washington, D.C. 20546	
U.S. Coast Guard (G-P-1/62) 400 Seventh Street, SW Washington, D.C. 20590	2	Commanding Officer Air Force Office of Scientific Research Technical Library Washington, D.C. 20301	1
National Aviation Facilities Experimental Center Library Atlantic City, NJ 08405	1	Commander Naval Air Systems Command - AIR 340F Attn: CDR C. Hutchins Washington, D.C. 20361	2
Dr. R.A. Cook Flight Systems Department - Q153 RAE, Farnborough Hants, ENGLAND	1	Chief ARI Field Unit P. O. Box 476 Ft. Rucker, AL 36362	1
OUSD&E (R&AT) (E&LS) CDR Paul R. Chatelier Room 3D129, The Pentagon Washington, D.C. 20301	1	Dr. J.M. Rolfe Research Branch Hq. RAF Support Command RAF Upwood Huntington, Cambs, ENGLAND	1
	1	Chief of Naval Operations OP-115 Attn: I. Shiff Washington, D.C. 20350	1
	1	Technical Library Naval Training Equipment Center Orlando, FL 32813	1
		Chief of Naval Operations OP-596 Washington, D.C. 20350	1

NAVTRAEEQUIPCEN IH-325

Distribution List (Cont'd)

Commander Naval Air Test Center, CT 176 Patuxent River, MD 20670	1 Dr. Thomas Longridge AFHRL/OTR Williams AFB, AZ 85224	1
Office of Deputy Chief of Naval Operations Manpower, Personnel and Training (OP-01) Washington, D.C. 20350	1 Commander Naval Air Systems Command, AIR-950D Technical Library Washington, D.C. 20361	1
Mr. Ken Staples Flight Research Division Royal Aircraft Establishment Bedform MK41 6AE, ENGLAND	1 Dr. David C. Nagel 1 LM-239-3 NASA Ames Research Center Moffett Field, CA 94035	1
Dr. Martin Tolcott Office of Naval Research 800 N. Quincy Street Department of the Navy Arlington, VA 22217	1 Federal Aviation Administration 1 Technical Library Bureau Research & Development Washington, D.C. 20590	1
Commander Naval Air Development Center Attn: Technical Library Warminster, PA 18974	1 Commander Naval Weapons Center 1 Human Factors Branch (Code 3194) Attn: Mr. Ronald A. Erickson China Lake, CA 93555	1
Naval Research Laboratory Attn: Library Washington, D.C. 20375	1 Lt Col Jefferson Koonce 1 USAFA/DFBL USAF Academy, CO 80840	1
Chief of Naval Education and Training Liaison Office AFHRL/OTLN Williams AFB, AZ 85224	6 CDR Joseph Funaro 2 Code 602 Human Factors Engineering Division Naval Air Development Center Warminster, PA 18972	1
Office of Naval Material MAT 08D2, CP5, Room 678 Attn: R. Fisher Washington, D.C. 20360	1 Mr. Don Gum AFWAL/FIGD Wright-Patterson AFB, OH 45433	1
Commander Officer Naval Education Training Program and Development Center Attn: Technical Library Pensacola, FL 32509	1 Mr. James Basinger 1 ASD/YWE Wright-Patterson AFB, OH 45433	2
Commander Pacific Missile Test Center Point Mugu, CA 93042	1 Mr. Rowland Bowles NASA Langley Research Center MS 125-B 1 Hampton, VA 23365	1
Commander Naval Air System Command, AIR 413 Washington, D.C. 20361	1 Mr. Brian Goldiez PM TRADE 1 DRCPM-PND-RE Naval Training Center Orlando, FL 32813	1

NAVTRAEEQUIPCEN IH-335

Distribution List (Cont'd)

Mr. David L. Key
U.S. Army AVRADCOM, M.S. 215-1
Ames Research Center
Moffett Field, CA 94035

1 Capt. A. Ginsburg
Aviation Vision Laboratory
AFAMRL/HEA
Wright-Patterson AFB, OH 45433

1

